US005850443A

United States Patent [19]

Van Oorschot et al.

[11] Patent Number: 5,850,443

[45] Date of Patent: *Dec. 15, 1998

[54] **KEY MANAGEMENT SYSTEM FOR MIXED-TRUST ENVIRONMENTS**

[75] Inventors: **Paul C. Van Oorschot**, Ottawa; **Michael James Wiener**, Nepean, both of Canada

[73] Assignee: **Entrust Technologies, Ltd.**, Ottawa, Canada

[*] Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

[21] Appl. No.: **698,074**

[22] Filed: **Aug. 15, 1996**

[51] **Int. Cl.**[6] **H04L 9/08**; H04L 9/30; H04L 9/00

[52] **U.S. Cl.** **380/21**; 380/9; 380/30; 380/45; 380/47; 380/49

[58] **Field of Search** 380/9, 21, 30, 380/44, 45, 46, 47, 49, 50, 20

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,962,539	6/1976	Ehrsam et al. .	
4,200,770	4/1980	Hellman et al. .	
4,405,829	9/1983	Rivest et al. .	
4,716,588	12/1987	Thompson et al.	380/20
4,882,779	11/1989	Rahtgen .	
4,914,697	4/1990	Dabbish et al.	380/49 X
5,199,069	3/1993	Barrett et al.	380/21 X
5,222,136	6/1993	Rasmussen et al. .	
5,416,841	5/1995	Merrick .	
5,511,123	4/1996	Adams	380/29
5,659,618	8/1997	Takahashi .	

OTHER PUBLICATIONS

Secure Hash Standard–FIPS PUB 180–1, U.S. Dept. of Commerce, Technology Administration, National Institute of Standards Technology, Apr. 17, 1995, pp. 1–21.

"A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", T. ElGamal, IEEE Transactions on Information Theory vol.IT–31, No.4, Jul. 1985, pp. 469–472.

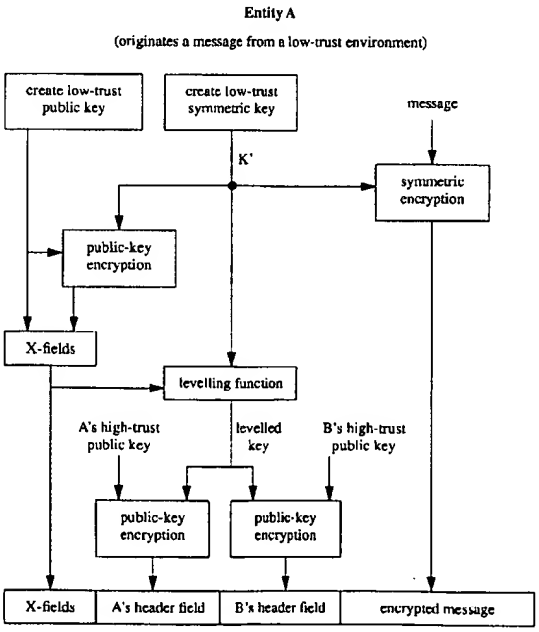
Primary Examiner—Bernarr E. Gregory

Attorney, Agent, or Firm—Markinson & Reckamp, P.C.

[57] **ABSTRACT**

The invention allows for transporting, in different degrees of security strength, a symmetric key encrypted using an asymmetric encryption technique, and along with this transporting ciphertext derived from plaintext encrypted under this symmetric key. The encryptor encrypts the plaintext using a symmetric whose strength is commensurate with the trust level of the environment in which the encryptor is located. The encryptor encrypts this symmetric key for one or more intended recipients using an asymmetric technique commensurate with a high-trust environment. In the case of the encryptor residing in the low-trust environment, the encryptor additionally encrypts this symmetric key using an asymmetric encryption public key of the originator itself (or alternatively, that of a third party). Decryption equipment in all environments uses the decryption process corresponding to an algorithm identifier included by the originator. In all cases, the asymmetric encryption/decryption process used for each specific recipient is of a strength commensurate with the trust level of that recipient's own environment.

21 Claims, 4 Drawing Sheets



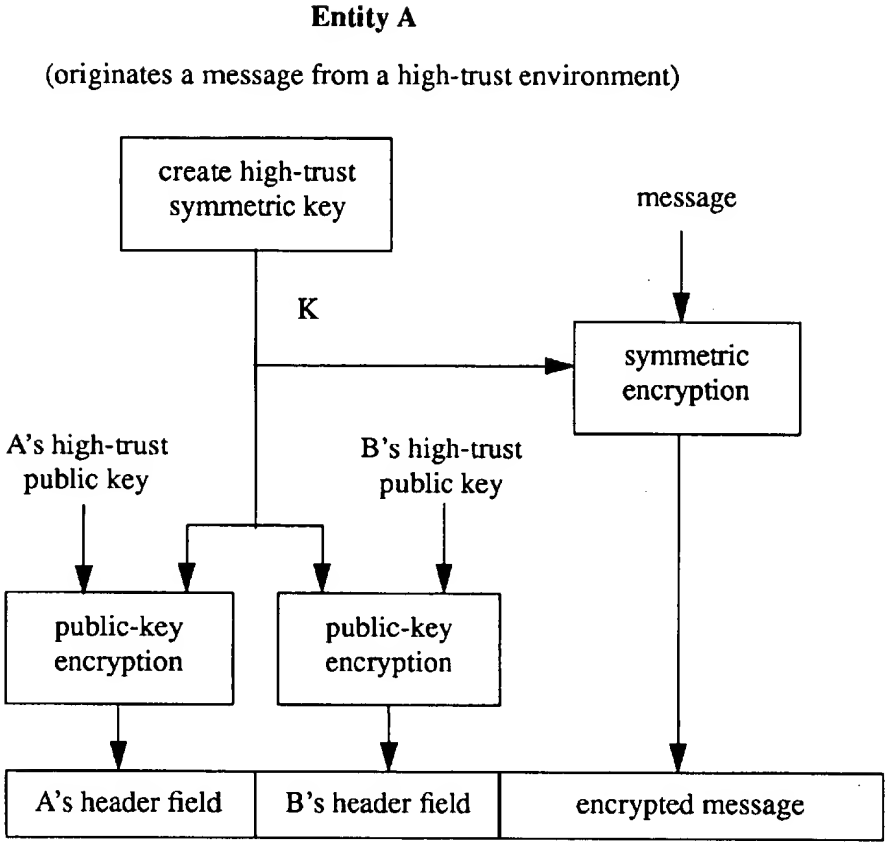

Entity A
(originates a message from a high-trust environment)
create high-trust symmetric key
message
K
symmetric encryption
A's high-trust public key
B's high-trust public key
public-key encryption
public-key encryption
A's header field
B's header field
encrypted message

Figure 1

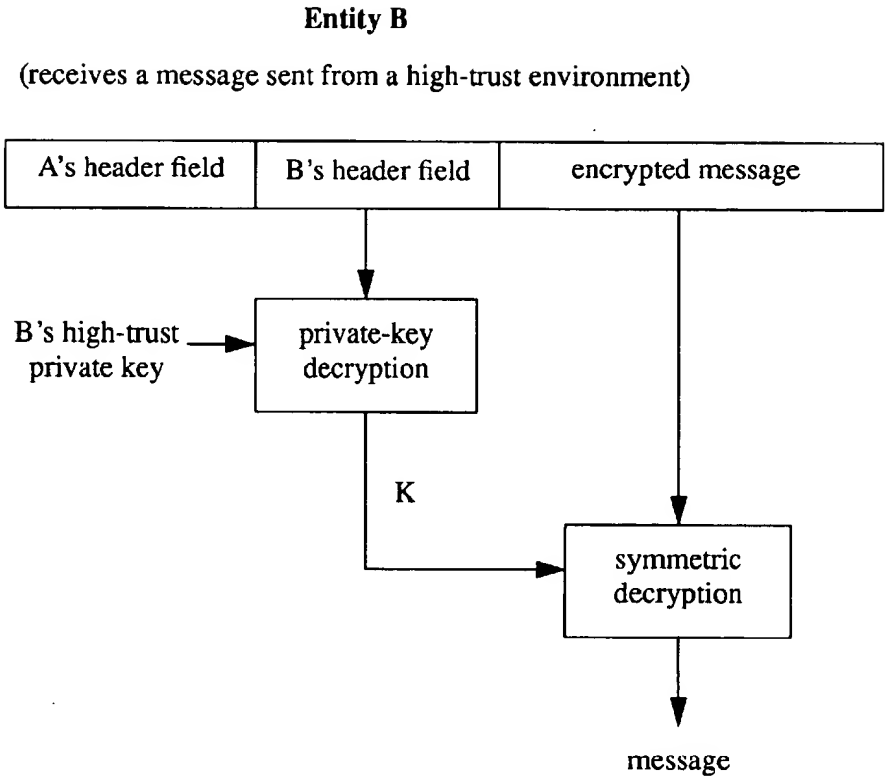

Entity B
(receives a message sent from a high-trust environment)
A's header field
B's header field
encrypted message
B's high-trust private key
private-key decryption
K
symmetric decryption
message

Figure 2

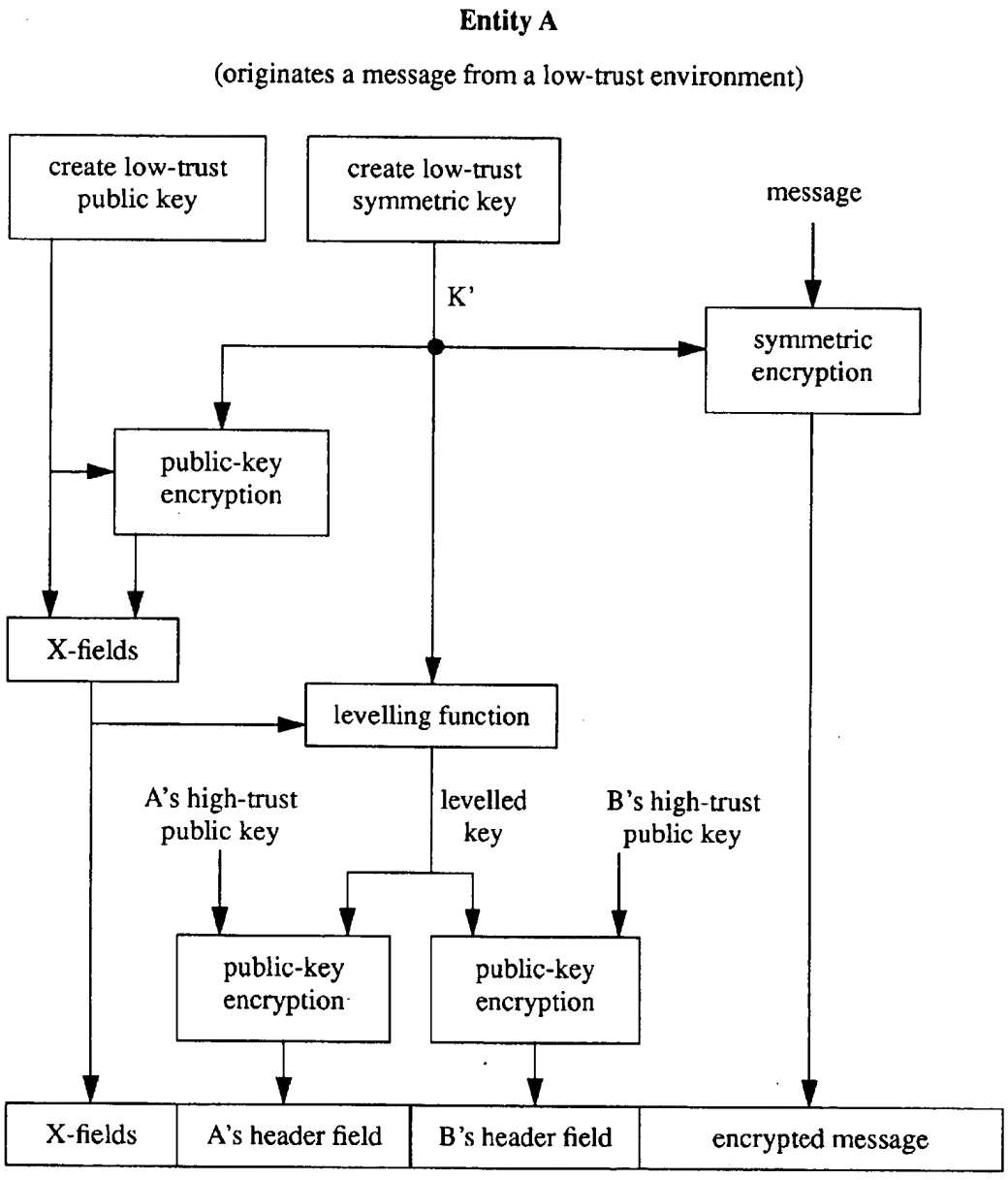

Entity A
(originates a message from a low-trust environment)
create low-trust public key
create low-trust symmetric key
message
K'
symmetric encryption
public-key encryption
X-fields
levelling function
A's high-trust public key
levelled key
B's high-trust public key
public-key encryption
public-key encryption
X-fields
A's header field
B's header field
encrypted message

Figure 3

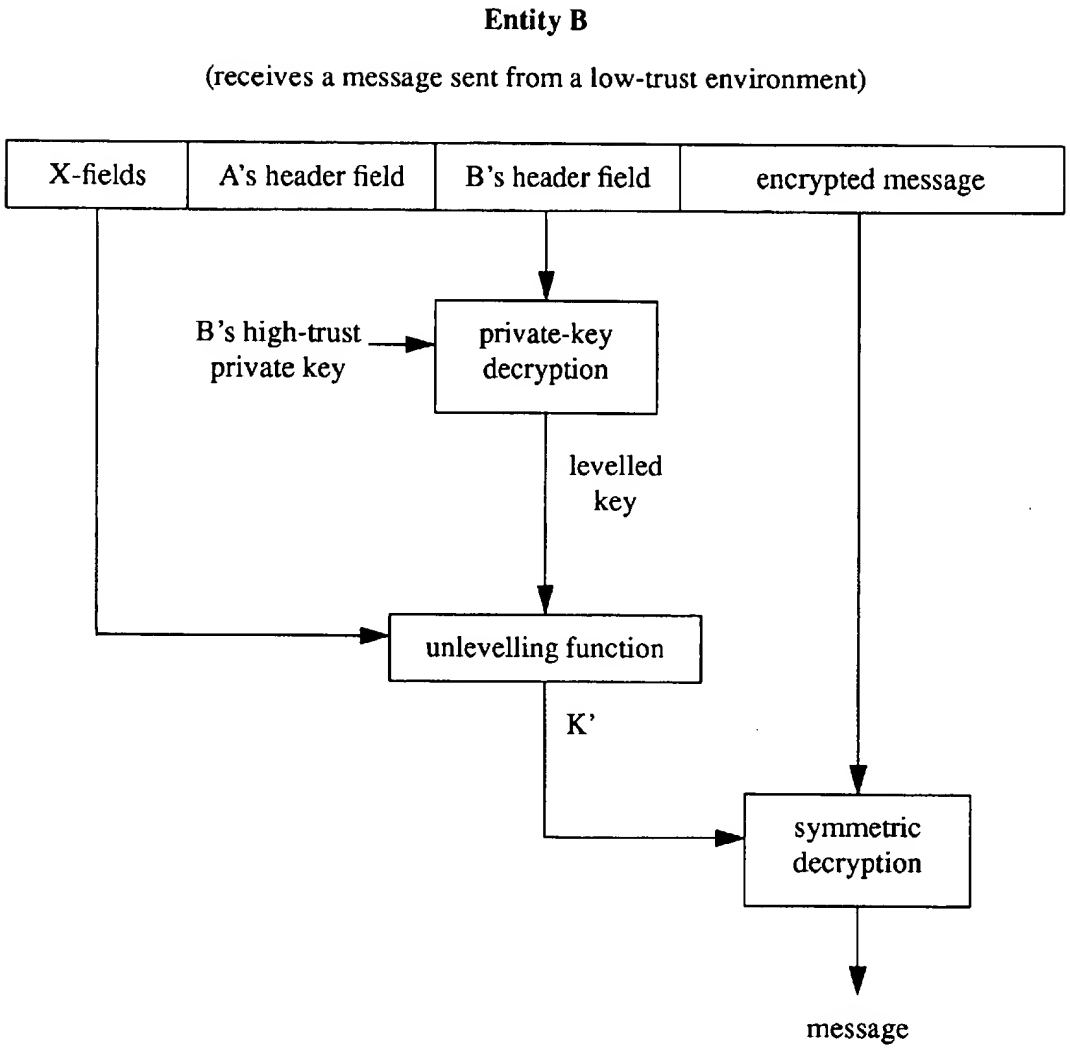

Entity B
(receives a message sent from a low-trust environment)
X-fields
A's header field
B's header field
encrypted message
B's high-trust private key
private-key decryption
levelled key
unlevelling function
K'
symmetric decryption
message

Figure 4

KEY MANAGEMENT SYSTEM FOR MIXED-TRUST ENVIRONMENTS

FIELD OF INVENTION

The invention resides generally in the field of key management for information security systems. More specifically, the invention relates to key management in communications networks which span environments of varying levels of trust.

BACKGROUND OF INVENTION

Encryption systems consist of an encryption process (or algorithm) and a decryption process. The input to the encryption process is a cryptographic key and data, which is referred to as plaintext data. The input to the decryption process is a cryptographic key and data, which is referred to as ciphertext data. The encryption process converts plaintext into ciphertext, while the decryption process does the converse. One characteristic of the key in an encryption system is its length, here denoted as k bits (a bit is a binary digit, representing a 0 or a 1).

In a symmetric encryption system, data to be protected, called plaintext, is encrypted in one environment to produce ciphertext. The ciphertext is decrypted in a second environment to recover the original plaintext. A number, called a key (or more specifically, a symmetric key) is shared between the encrypting and the decrypting process. The key must be secret, but the ciphertext encrypted under this key can be transmitted over an otherwise unprotected communications medium which is subject to eavesdropping by an adversary. The adversary is unable to recover the plaintext due to lack of knowledge of the key. In well-designed symmetric encryption systems, all k bits of a key are necessary for the encryption and decryption algorithms to function properly. Examples of symmetric encryption algorithms are the Data Encryption Standard (DES), originally detailed by Ehrsam et al. in U.S. Pat. No. 3,962,539; block ciphers constructed using the CAST design technique of Adams, details of which are given in U.S. Pat. No. 5,511,123 Apr. 26, 1996; and well known proprietary block ciphers such as the RC2 cipher of RSA Data Security Inc..

Cryptographic techniques other than encryption also make use of symmetric keys. One example is message authentication code (MAC) algorithms, which involve appending to a transmitted message a tag value (or MAC), which is computed using an algorithm which takes as input the message data and a secret key. The recipient, who shares the secret key, upon receiving the data and tag recomputes its own tag value from the shared key and the received data, and compares this tag value to that received. If the tag values agree, the recipient has some assurance that the data originated from the party with which it shares the key. MACs thus provide data origin authentication.

Symmetric encryption algorithms may be attacked by an adversary who, given one known plaintext-ciphertext pair of data, tries all 2^k possible k-bit keys to see which one maps the known plaintext to the known ciphertext. This is referred to as an exhaustive key search. In a well-designed symmetric encryption system, an adversary can do no better than mount such an exhaustive attack. In this case, the bit-length k of the key gives an indication of the strength of the algorithm, the work required for an attack is 2^k operations, and the probability of any particular key being guessed, assuming that all keys are equi-probable, is $(\frac{1}{2})^k$.

Asymmetric cryptographic techniques, such as the RSA scheme of Rivest, Shamir and Adleman of U.S. Pat. No. 4,405,829, also play a major role in commercial cryptographic solutions in the field of information security. The basic idea is as follows. An encryption algorithm, for example, is parameterized by a pair of related numbers, known as a private key and a public key. The public key, known to everyone, allows anyone to encrypt data for a specific intended recipient; the private key, known only to the intended recipient, allows only that individual to decrypt the data. Another asymmetric technique, referred to as DH key exchange after Diffie and Hellman, and described by Hellman, Diffie and Merkle in U.S. Pat. No. 4,200,770, allows two parties to establish a shared secret key using only publicly known parameters. DH can also be used for key transfer to provide functionality equivalent to RSA key transfer; this is commonly called ElGamal encryption (see T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Transactions on Information Theory volume 31, 1985, pages 469–472). Variations of ElGamal encryption have also been proposed and implemented using elliptic curve cryptography.

In practice, asymmetric techniques are often used for key management applications, and in particular, for the transfer of a symmetric key from one party to one or more other parties. Often a different symmetric key is used for each transmission from a party A to a party B; in this case, the symmetric key is referred to as a session key. The session key is then typically used in a symmetric algorithm, e.g. an encryption algorithm such as DES or a CAST algorithm. This is done because symmetric encryption algorithms are often faster for bulk data encryption than asymmetric techniques, while the latter allow for more convenient solutions to the key distribution problem because only the authenticity of a public key need be assured, and this is easier than distributing keys whose secrecy must be guaranteed. Such systems involving both symmetric and asymmetric techniques are called hybrid systems.

A common example of a hybrid technique is to encrypt a data file with a symmetric key to produce ciphertext, and to format this ciphertext as a data file with a header. The header contains one or more copies of the symmetric key, encrypted using the public key of one or more intended recipients. The key asymmetrically encrypted for each recipient is preceded by an identifier which allows the intended recipient to determine which of the possibly multiple fields in the header is the one appropriate for it to decrypt in order to recover the symmetric key. This technique is referred to as digital enveloping.

When cryptographic techniques are used in communications systems which span different (e.g., geographic) regions, in practice it may occur that the different regions can be considered to be trusted to different extents. For example, region X may be considered a high-trust environment because it lies entirely within a country having no concerns about unlawful use of encryption, e.g. because the laws of that country allow law-enforcement access to encryption keys under appropriate circumstances (e.g. wiretaps authorized by one or more judges or other trusted agents). In contrast, a region Y may be considered a low-trust environment because there is some risk that within it, encryption may be used for purposes which subvert law-enforcement or the protection of national security, or because appropriate legislative or administrative safeguards are not in place.

The usual approach (hereafter called the lowest-level approach) to using cryptographic techniques in such mixed-trust environments is to have both a strong and a weak cryptographic technique. Products installed in the low-trust

environment are restricted to containing only the weak algorithm, while those in the high-trust environment contain both the strong and weak techniques. By this approach, communications in which both end-points reside in the strong environment may provide security using the strong techniques, whereas for reasons of interoperability, communications in which one or both end-points reside in the low-trust environment can be protected only by the weak techniques. This allows authorities to intercept communications involving the low-trust environment and defeat the cryptographic protection if necessary for national security or law enforcement reasons.

A notable exception to the prior-art lowest-level approach is the mixed-trust encryption system of Ford, specified in the co-pending U.S. patent application Ser. No. 08/535,445 filed on Sep. 28, 1995 now allowed and assigned to the assignee of the present invention. That invention provides a solution to the mixed-trust use of a symmetric encryption algorithm, while the focus of the present invention is key management in a mixed-trust environment, and including mixed-trust key management using asymmetric techniques. The present invention provides a mixed-trust key management solution which is complementary to the invention of application Ser. No. 08/535,445.

The lowest-level approach has at least two drawbacks, which apply for both the case that the cryptographic technique in question is a symmetric encryption algorithm used for bulk encryption as per application Ser. No. 08/535,445 and when an asymmetric cryptographic technique is used for key establishment as per the present invention. The first drawback is that the lowest-level approach unnecessarily degrades the security of the system when communications originating in the high-trust environment are destined for recipients in both the low-trust environment and the high-trust environment (or a low-trust environment alone), because in this case the approach makes the communications susceptible to an adversary capable of defeating the weaker technique. The present invention overcomes this deficiency, while maintaining the objective of guarding against entities in the low-trust environment from using high-trust cryptographic key management techniques for purposes which may subvert law-enforcement or the protection of national security.

The second drawback of the lowest-level approach is that it unnecessarily increases the complexity of products in the high-trust environment, by requiring such products which originate communications from knowing, at the time a communication is originated, whether the intended recipient (s) are in the high-trust environment or the low-trust environment. In some cases, this constraint may even preclude deployment of a product, if the system architecture is unable to make such information available to the originator. The present invention removes this deficiency, such that an originating entity in the high-trust environment performs the same key management process regardless of the trust-level of the environment of the intended recipient(s). Likewise, originating entities in low-trust environments carry out the same operation regardless of the environment of their intended recipient(s). Receiving entities in both high-trust and low-trust environments are able to carry out the appropriate reception operations based on identifying information included by the originator in the transmitted message.

OBJECTS OF INVENTION

It is therefore an object of the present invention to provide a method and a system for establishing shared secret cryptographic keys between two or more parties over a communication network which spans both high-trust and low-trust environments.

It is another objective of the present invention to ensure a secure data transfer which originates in the high-trust environment and for which the intended recipients are either in the high-trust environment or the low-trust environment.

It is another object that entities in the high-trust environment need not carry out any special operations which might otherwise be required to distinguish incoming communications originating other high-trust environment from those which originated in the low-trust environment.

SUMMARY OF INVENTION

Briefly stated according to one aspect the invention is directed to a method of managing cryptographic keys between a first and second parties in communication environments of different degrees of trust. The method comprises steps of the first party encrypting a cryptographic key of a cryptographic strength commensurate with the degree of trust of the environment in which the first party is located, by using a low trust encryption public key of the first party to generate a first party encrypted cryptographic key. The first party separately encrypts the cryptographic key using a high trust encryption public key of the second party to generate a second party encrypted cryptographic key, and concatenates the first and second encrypted cryptographic keys. The method further includes a step of the second party, upon reception of the concatenated data, decrypting the second encrypted cryptographic key to recover the cryptographic key.

According to another aspect, the invention is directed to a method of managing cryptographic keys between a first and second parties in communication environments of different degrees of trust. The method comprises steps of the first party selecting a cryptographic key of a cryptographic strength commensurate with the degree of trust of the environment in which the first party is located and performing a levelling function involving combining, using a reversible function, the cryptographic key with additional data derived in part or in whole from the data field described below, to generate a levelled key. The method further includes steps of the first party encrypting the levelled key using a high trust encryption public key of the second party to generate a second party encrypted levelled key. The method includes a further step of the first party creating a data field consisting in part of the cryptographic key, encrypted under a low trust encryption public key of the first party and concatenating the data field and second party encrypted levelled key. The method yet includes steps of the second party, upon reception of the concatenated data, decrypting the second party encrypted levelled key to recover the levelled key, and performing an unlevelling function, using the received data field and the recovered levelled key to recover the cryptographic key.

BRIEF DESCRIPTION OF DRAWINGS

FIGS. **1**, **2**, **3** and **4** are illustrative examples of algorithmic processes of an encryptor and a decryptor supporting the method according to embodiments of the invention.

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS OF INVENTION

According to one aspect, the invention resides in a mechanism and supporting system whose design allows for

transporting, in different degrees of security strength, a symmetric key encrypted using an asymmetric encryption technique, and optionally along with this transporting ciphertext derived from plaintext encrypted under this symmetric key. The method includes the encryptor encrypting the plaintext using a symmetric encryption process whose strength is commensurate with the trust level of the environment in which the encryptor is located, using a symmetric key of a corresponding strength; using, for transmissions originating in both the low-trust and the high-trust environment, an asymmetric technique commensurate with a high-trust environment to encrypt this symmetric key for one or more intended recipients; and in the case of the encryptor residing in the low-trust environment, additionally encrypting this symmetric key using an asymmetric technique commensurate in strength with the low-trust environment using an asymmetric encryption public key of the originator itself (or alternatively, that of a third party) referred to as key X below. The encryption under key X effectively reduces the overall security to that of the low-trust environment in, and only in, the special case where the originating equipment resides in the low trust environment. Decryption equipment in all environments uses the decryption process corresponding to an algorithm identifier included by the originator. In all cases, the asymmetric encryption/decryption process used for each specific recipient is of a strength commensurate with the trust level of that recipient's own environment. Furthermore, in the case that the originator is in a low-trust environment, the data recovered by asymmetric decryption by the recipient must be combined with a data value which is some function of the ciphertext encrypted under key X in order to recover the symmetric key which allows the recipient to recover the original plaintext. This feature guarantees that the presence of the data field associated with key X cannot be removed in order to, contrary to the design intent, "upgrade" the trust-level of the low-trust equipment, because doing so prevents recipient equipment from recovery of the intended data.

According to another aspect, the invention is directed to an apparatus for complementary cryptographic operations, such as encryption and decryption, in different degrees of security strength. The apparatus comprises either one or both of a first symmetric encryption module for use in encrypting data in high-trust environments which uses a strong cryptographic process, and a second encryption module for use in encrypting data in low-trust environments which uses a less strong symmetric cryptographic process; together with one or both of a first asymmetric encryption/decryption module for use in key transfer providing a security strength commensurate with a high-trust level environment, and a second asymmetric encryption/decryption module for use in key transfer providing a security strength commensurate with a low-trust environment; and finally, also includes a module providing a mechanism capable of determining the source of received cryptographically protected information, allowing a decision to be made to allow proper recovery of an asymmetrically-encrypted symmetric key to allow such key to be used to decrypt symmetrically-encrypted plaintext data.

Reference is now made to FIGS. **1** and **2**. In one embodiment, the invention involves use of the RSA public-key encryption technique for key transfer from one party to one or more parties over an otherwise unsecured communications channel, and using the digital enveloping technique described above. The plaintext data file is encrypted once, e.g. using the DES or a CAST symmetric algorithm, and a new random symmetric key (referred to below as the file key). The RSA public key of each intended recipient is obtained by the originator using some means which guarantees the authenticity of the key. Each public key is then used to encrypt a separate copy of the file key. The copies of the file key are then included in a file header, followed by one copy of the encrypted data itself.

More specifically, one preferred embodiment of the invention involves the following components. The low-trust system module is constrained to use 512-bit RSA encryption for key transfer, while the high-trust system makes use of 1024-bit RSA for key transfer. Following the invention disclosed in U.S. patent application Ser. No. 08/535,445, the low-trust system is designed to decrypt data files using 80-bit keys, and to encrypt data files using 40-bit keys; this is called an "80/40 export solution". Despite the 512-bit constraint on the low-trust environment, all entities in the communications system have 1024-bit RSA public encryption keys which are made available to other system entities, e.g. through a public directory. Entities which reside in the low-trust environment have, in addition, a 512-bit RSA encryption public key which need not be used by any other entities, and therefore need not appear in the directory; in fact, these 512-bit keys may optionally be generated on a per-use basis for each communication.

If entities A and B are both in a high-trust environment, and A wishes to send a data file to B, A (i.e. the cryptographic module of the equipment which user A is using) symmetrically encrypts the data file using a new 80-bit CAST key K, and then RSA-encrypts one copy of K under its own 1024-bit RSA key, and a second copy of K under the 1024-bit RSA key of B. The two encrypted keys are included in the header of a file which also includes the encrypted data file. The composite file is then sent to B.

In the case that B resides in a low-trust environment, the cryptomodule of entity A generates the same composite file, and sends this to B.

Referring now to FIGS. **3** and **4**, in the case that A resides in a low-trust environment, and is communicating with an entity B which resides in either a low-trust environment or a high-trust environment, A's equipment generates the following data instead. A 40-bit CAST key K' is used to symmetrically encrypt the data file, some function of K' (called the levelled key) is RSA-encrypted under each of the 1024-bit keys as above, and two additional data fields are included in the file header. Regarding these additional fields, the first is a 512-bit RSA public key of entity A itself, and the second is the RSA-encryption of K' under this 512-bit key. Let X denote the concatenation of these two data fields, and let h(X) denote the "hash" of the data string X, e.g. using a one-way hash function such as the Secure Hash Algorithm SHA-1 as specified in U.S. Federal Information Processing Standards Publication 180-1 (FIPS PUB 180-1). Alternatively, another well known MD5 hash function or RIPEMD-160 can be used.

The function of K' (the levelled key) which is RSA-encrypted rather than the 80-bit key is (K' XOR h40(X)), where XOR is the bitwise exclusive-OR operation, and h40(X) denotes 40 bits, say the leftmost 40 bits, of the value h(X). The use of the levelled key, rather the K' alone, is one means to ensure that the fields which compose X are not simply removed by a party which wishes to "upgrade" the overall security of the communication to a 1024-bit RSA encryption (as is the case earlier where both A and B resided in the high-trust environment). Thus the following fields are transmitted from A to B in the case that A is in a low-trust environment, and B is either in a low-trust or high-trust

environment: X, RSA1024_A(K' XOR h40(X)), RSA1024_B(K' XOR h40(X)), CAST40 (data file). Here K' is a 40-bit symmetric CAST key, CAST40(–) denotes symmetric encryption of the bracketed quantity using a 40-bit symmetric CAST algorithm, and X is the concatenation of: a 512-bit RSA public key of A, and K' RSA-encrypted under this key.

While the described embodiment involves the use of 512-bit and 1024-bit RSA, 40-bit and 80-bit CAST, the particular hash function SHA-1, and a levelled key created by the XOR of two quantities, the invention can clearly be modified for different asymmetric keys lengths and different public-key encryption techniques, different symmetric key lengths and different symmetric key algorithms, different hash functions, and different key-levelling functions. These can all be varied to match different trust level requirements of different environments and systems, and the algorithms preferred for use in different systems.

In the case that entity A in the low trust environment is communicating with both entity B (which uses 1024-bit RSA keys) and some other entity C (which uses 512-bit RSA keys), no special access modifications are made for entity C. The header field for entity C would consist of the 40-bit key K' encrypted with C's 512-bit RSA key. Because entity C uses an RSA key size consistent with a low trust environment, no levelling operations are required. In this way, interoperability is maintained with entities which use low trust RSA key sizes and do not support levelling functionality.

As seen in the above discussion, the present invention provides a method and a system for establishing shared secret keys (e.g. to allow encryption and/or other cryptographic protection including authentication), between two or more parties over a communication network which spans both high-trust and low-trust environments.

The present invention also ensures that cryptographic keys, used for cryptographic protection of data in high-trust environments, are not unnecessarily exposed (i.e. downgraded to a reduced-trust level) to eavesdroppers or adversaries when such keys and the data protected thereunder are transmitted in a key establishment communication and data transfer which originates in the high-trust environment and for which the intended recipients are either in the high-trust environment or the low-trust environment.

The invention provides an apparatus and system design such that equipment in the high-trust environment which is the source of the cryptographically protected information or key transfer, need not know at the time of transfer whether the protected information or key is destined for a high-trust or a low-trust environment.

According to the invention, entities in the high-trust environment, upon receiving cryptographically protected communications from other entities in the high-trust environment, need not carry out any special operations which might otherwise be required to distinguish such incoming communications from those which had originated in the low-trust environment; and that for incoming communications originating in the low-trust environment, the high-trust recipient carries out operations which enforce the requirement that the cryptographic protection used by the low-trust originator was indeed that (and no higher than that) which was designed into the system architecture.

The invention also ensures that persons using equipment incorporating the method and system of the present invention in the low-trust environment are unable to subvert the intended design feature, ensuring that entities be unable to originate messages with cryptographic protection at the same level of security as that provided by the corresponding high-trust environment equipment, and which might therefore subvert the design features supporting law-enforcement actions.

What is claimed is:

1. A method of managing cryptographic keys between first and second parties in communication environments of different degrees of trust comprising the steps of:

the first party

encrypting a cryptographic key by using a low trust encryption public key of the first party having a first key length, to generate a first party encrypted cryptographic key,

encrypting the cryptographic key using a higher trust encryption public key of the second party having a second key length longer than the first key length to generate a second party encrypted cryptographic key, and

concatenating the first party and second party encrypted cryptographic keys, and

the second party, upon reception of the concatenated data, decrypting the second party encrypted cryptographic key to recover the cryptographic key.

2. The method according to claim 1, wherein the cryptographic key is an encryption key, and comprising further steps of:

the first party

encrypting plaintext into ciphertext using the cryptographic key,

concatenating the ciphertext to the first party and second party encrypted cryptographic keys;

the second party

decrypting the ciphertext into the plaintext using the thus recovered cryptographic key.

3. The method according to claim 2 wherein the cryptographic key is a symmetric encryption key and the first and second parties use a symmetric encryption process for encrypting the plaintext or decrypting the ciphertext.

4. The method according to claim 3 wherein the symmetric encryption process is a block cipher from the group of DES, CAST and RC2.

5. The method according to claim 1 wherein the first and second parties use distinct asymmetric encryption processes to generate the first party and second party encrypted cryptographic keys.

6. The method according to claim 5 wherein the asymmetric encryption processes are any of RSA encryption and ElGamal encryption.

7. The method according to claim 1 wherein there are three or more parties in communication environments of different degrees of trust comprising steps of:

for the third, and other remaining parties separately encrypting the cryptographic key using an encryption public key of each of these parties to generate a second party, third party and additional encrypted cryptographic keys, and

concatenating the first, second, and additional encrypted cryptographic keys,

the second and subsequent parties each, upon reception of the concatenated data,

decrypting the corresponding encrypted cryptographic key to recover the cryptographic key.

8. A method of managing cryptographic keys between first and second parties in communication environments of different degrees of trust comprising the steps of:

the first party

selecting a cryptographic key,

creating a data field consisting in part of the cryptographic key, encrypted under a low trust encryption public key of the first party having a first key length,

combining, using a reversible function, the cryptographic key with additional data derived in part or in whole from the data field to generate a levelled key,

encrypting the levelled key using a high trust encryption public key of the second party having a second key length longer than the first key length to generate a second party encrypted levelled key,

concatenating the data field, and second party encrypted levelled key,

the second party, upon reception of the concatenated data,

decrypting the second party encrypted levelled key to recover the levelled key, and

recovering the cryptographic key using the received data field and the recovered levelled key.

9. The method according to claim **8**, wherein the cryptographic key is an encryption key, and comprising further steps of:

the first party

encrypting a plaintext into a ciphertext using the cryptographic key, concatenating the ciphertext to the data field and the second party encrypted levelled key,

the second party, upon reception of the concatenated data, decrypting the ciphertext into the plaintext using the thus recovered cryptographic key.

10. The method according to claim **9** wherein the cryptographic key is a symmetric encryption key and the first and second parties use a symmetric encryption process for encrypting the plaintext or decrypting the ciphertext.

11. The method according to claim **10** wherein the symmetric encryption process is a block cipher from the group of DES, CAST and RC2.

12. The method according to claim **8** wherein the first party uses distinct asymmetric encryption processes to generate the second party encrypted levelled key and the second party uses an asymmetric decryption process to decrypt the second party encrytped levelled key.

13. The method according to claim **12** wherein the asymmetric encryption processes are any of RSA encryption, and ElGamal encryption.

14. The method according to claim **8** wherein the step of combining using a reversible process to generate a levelled key comprises further steps of:

encrypting the cryptographic key using the low trust encryption public key of the first party having the first key length,

concatenating the resulting data to said low trust encryption public key itself,

hashing a resulting data string using a cryptographic hash function, resulting in a hash value,

combining a subset of the hash value, using an exclusive-OR operation, with said cryptographic key, to generate the levelled key.

15. The method according to claim **14** where the hash function used is from the group of SHA-1 and MD5, hash functions.

16. The method according to claim **8** wherein there are three or more parties in communication environments of different degrees of trust, comprising steps of:

for third, and other remaining parties separately encrypting the levelled key using an encryption public key of each of these parties to generate a third party and additional encrypted levelled keys, and

concatenating the second party, third party, and additional encrypted levelled keys,

the second and subsequent parities each, upon reception of the concatenated data,

decrypting the corresponding encrypted levelled key, and recovering the corresponding cryptographic key using the decrypted levelled key.

17. The method according to claim **8** wherein the data field consists of a low trust encryption public key of the first party having a key length shorter than a key length of a high trust encryption public key, concatenated to the encrypted value of the cryptographic key under the low trust encryption public key.

18. The method according to claim **8** comprising further steps of the first party

encrypting the levelled key by using a high trust encryption public key of the first party having a key length larger than the low trust encryption public key to generate a first party encrypted levelled key, and inserting the first party encrypted levelled key into the concatenated data.

19. An apparatus for complementary cryptographic operations, in different degrees of security strength comprising:

first encryption means for encrypting a cryptographic key by using a low trust encryption public key of the first party having a first key length, to generate a first party encrypted cryptographic key,

second encryption means for encrypting the cryptographic key using a higher trust encryption public key of the second party having a second key length longer than the first key length to generate a second party encrypted cryptographic key, and

means, responsive to the first and second encryption means, for concatenating the first party and second party encrypted cryptographic keys, and

means, responsive to the concatenated data, for decrypting the second party encrypted cryptographic key to recover the cryptographic key.

20. A method of managing cryptographic keys between first and second parties in communication environments of different degrees of trust comprising the steps of:

the first party

selecting a cryptographic key,

creating a data field consisting in part of the cryptographic key, encrypted under a low trust encryption public key of the first party having a first key length,

combining, using a reversible function, the cryptographic key with additional data derived in part or in whole from the data field to generate a levelled key,

encrypting the levelled key using a high trust encryption public key of the second party having a second key length longer than the first key length to generate a second party encrypted levelled key,

concatenating the data field, and second party encrypted levelled key,

the second party, upon reception of the concatenated data,

decrypting the second party encrypted levelled key to recover the levelled key, and

recovering the cryptographic key using the received data field and the recovered levelled key.

21. An apparatus for complementary cryptographic operations in different degrees of security strength comprising:

first encryption means for encrypting a cryptographic key by using a low trust encryption public key of the first party having a first key length, to generate a first party encrypted cryptographic key,

second encryption means for encrypting the cryptographic key using a higher trust encryption public key of the second party having a second key length longer than the first key length to generate a second party encrypted cryptographic key, and

means, responsive to the first and second encryption means, for concatenating the first party and second party encrypted cryptographic keys, and

means, responsive to the concatenated data, for decrypting the second party encrypted cryptographic key to recover the cryptographic key.

* * * * *

US005475757A

United States Patent [19]

Kelly

[11] Patent Number: 5,475,757

[45] Date of Patent: Dec. 12, 1995

[54] **SECURE DATA TRANSMISSION METHOD**

[75] Inventor: **Joseph P. Kelly**, Howell, N.J.

[73] Assignee: **AT&T Corp.**, Murray Hill, N.J.

[21] Appl. No.: **255,207**

[22] Filed: **Jun. 7, 1994**

[51] **Int. Cl.**[6] **H04K 1/00**

[52] **U.S. Cl.** **380/24**; 380/21; 380/23

[58] **Field of Search** 380/21, 23, 25, 380/24, 4, 49

[56] **References Cited**

U.S. PATENT DOCUMENTS

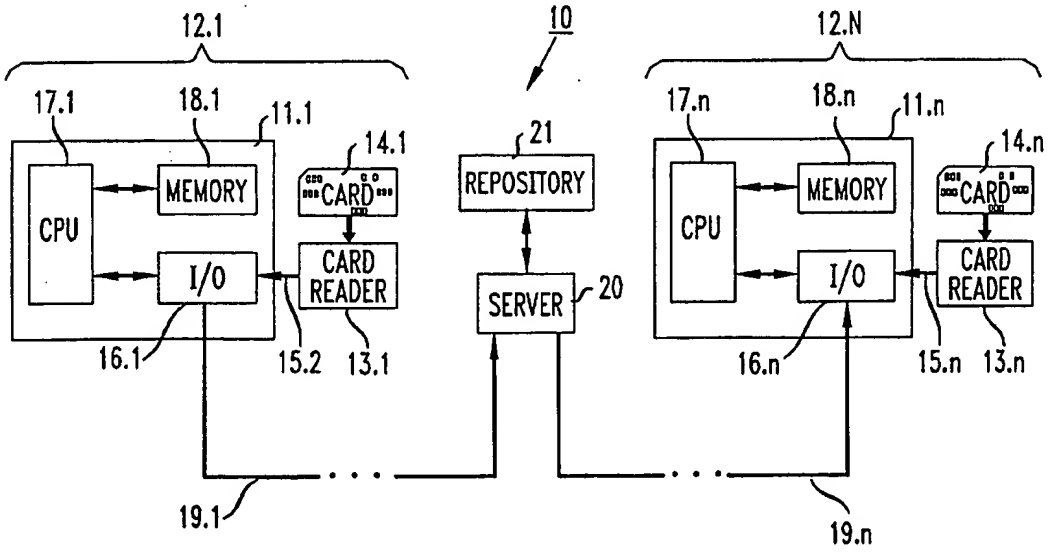
5,202,921	4/1993	Herzberg et al.	380/23
5,371,794	12/1994	Diffie et al.	380/21

Primary Examiner—David C. Cain

[57] **ABSTRACT**

A method of secure data transmission commences at a sending or originating terminal by processing a sender challenge on an originating subscriber card with a secret originating subscriber coding key to obtain an originating subscriber response. The response is used at the originating terminal to encrypt the message to be securely transmitted. The thus-encrypted message is transmitted, together with the sender challenge in its original form, to a system server. The server retrieves the originating subscriber coding key from a repository to which it has access, and uses the key to generate a response that is identical to that produced by the originating subscriber. It then employs the so-obtained response to decrypt the originator's encrypted message, determines the intended recipient, and retrieves from the repository the coding key assigned to such recipient. The server then issues a new challenge and repeats the above processing and encryption steps using the recipient's subscriber's coding key, thereby re-encoding the message and sending the thus re-encrypted message and the unencrypted new challenge to the receiving station where the same process is employed on the recipient subscriber card to obtain, from the unencrypted new challenge and the recipient subscriber code key stored on the recipient subscriber card, the receiving subscriber response to be used in decryption of the received, server re-encrypted message.

7 Claims, 2 Drawing Sheets



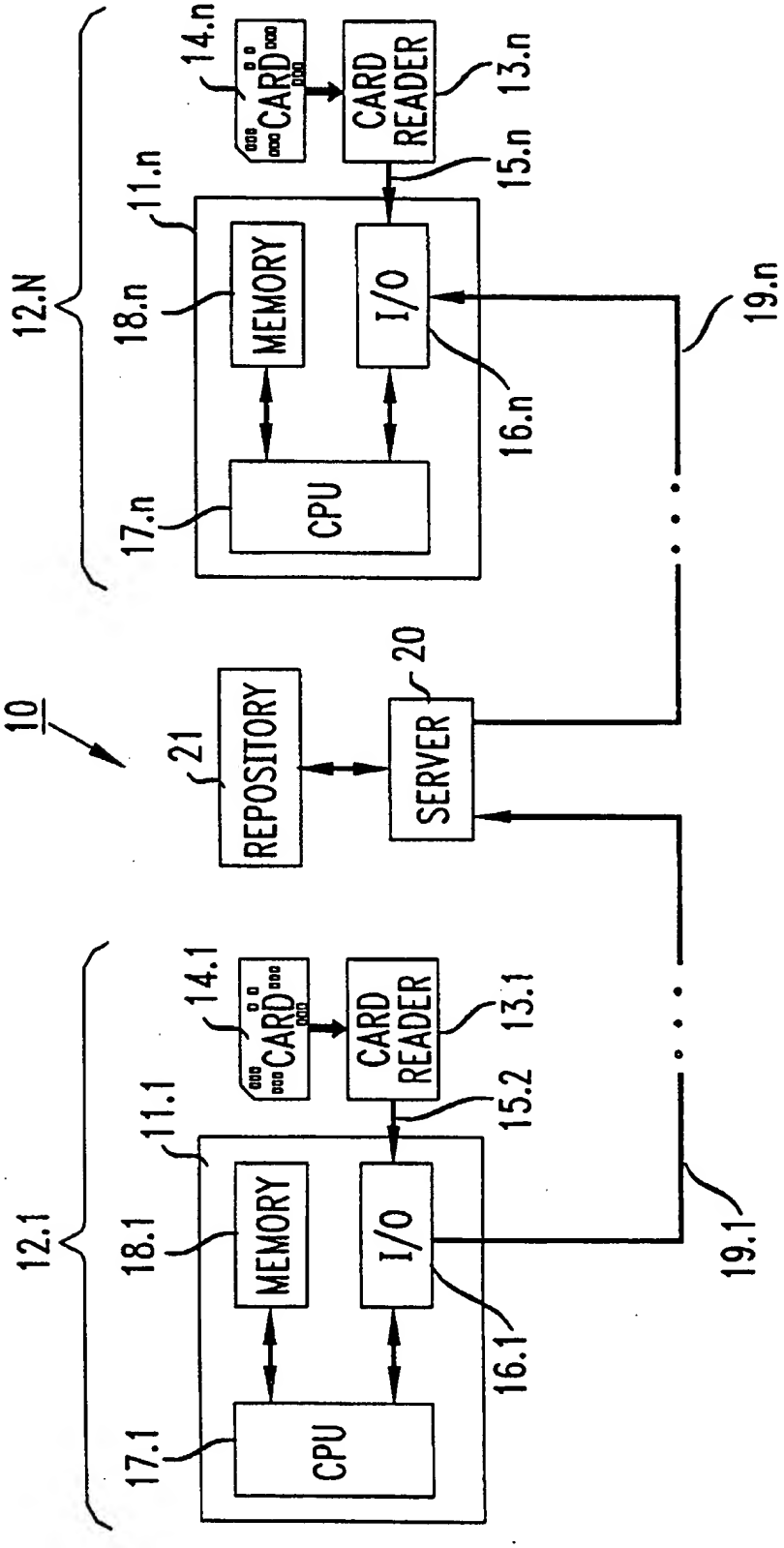
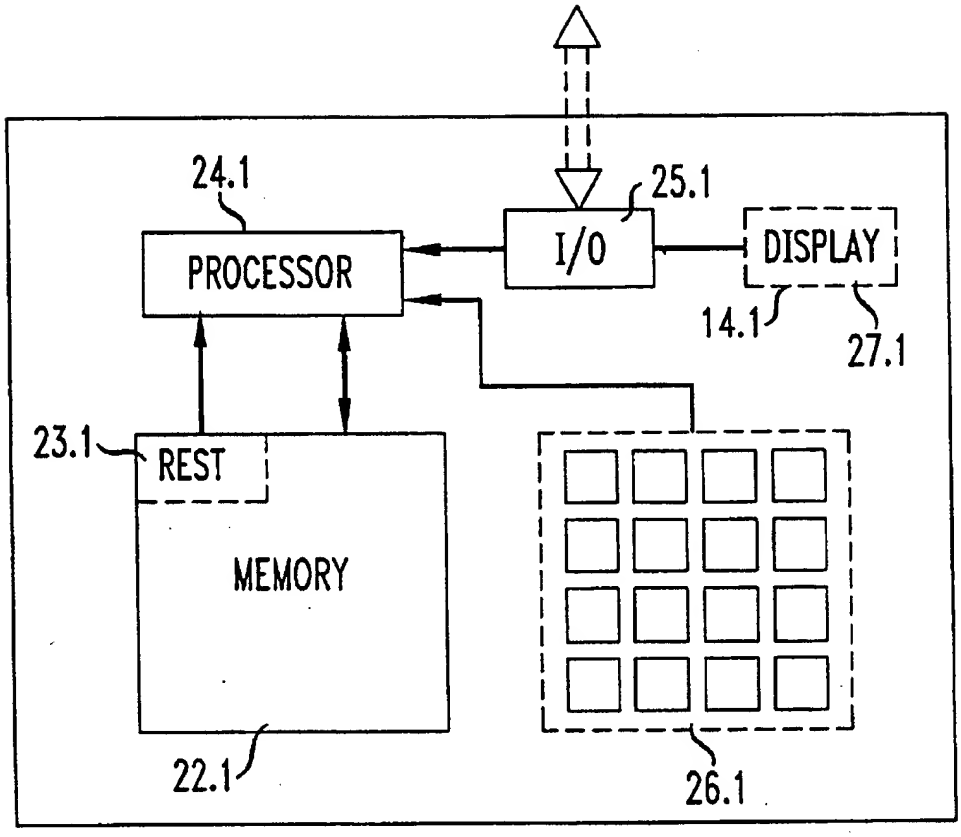

10
12.N
12.1
11.n
11.1
18.n
18.1
17.n
17.1
MEMORY
CPU
I/O
CARD
CARD READER
14.n
13.n
15.n
16.n
14.1
13.1
15.2
16.1
19.n
19.1
20
SERVER
21
REPOSITORY

FIG. 1

FIG. 2



SECURE DATA TRANSMISSION METHOD

FIELD OF THE INVENTION

The present invention relates to the telecommunications field in general, and more particularly to the area of encryption and decryption of electronically transmitted messages.

BACKGROUND ART

As electronic mail and other data transmission methods gain in popularity and become more and more widespread both as to the number of people availing themselves of such services and the number of messages being sent by way of these channels, there is a growing need for simple, secure and reliable encryption of the data being transmitted. This is especially so because a steadily increasing proportion of such data is of a proprietary or otherwise sensitive nature which, were such information to fall into the wrong hands, could be detrimental or, at the very least, embarrassing to the issuer of the transmitted information and/or its intended recipient.

To satisfy this need for transmission security, there have been developed and are currently available a variety of devices and algorithms for encoding information to be transmitted and for subsequently decoding the encoded information after it has reached the intended recipient. Of course, it is important to encrypt the information in such a manner as to make it difficult, if not substantially impossible, to break the code or key used in the encryption. This, coupled with safeguarding of the key itself by all persons having access to that key, provides a high degree of assurance that anyone who may have received or intercepted the transmission without being authorized to learn of its contents will be unable to decipher the message contained in the transmission.

Of course, it would be possible, and is in fact required when operating in accordance with the Digital Encryption Standard (DES) currently applicable to electronic mail (E-mail), for the issuer and the intended recipient to agree upon or notify one another of a particular encoding key to be used. Such key could then be employed for all encrypted communications sent by the respective issuer, or all those taking place between the respective parties, or such communications occurring within a certain time span, as on a particular day, or even to individual messages. This encoding key information exchange may take place either well in advance of the time for a particular communication or, especially when using a different key for each transmission, just prior to the intended transmission time.

Each of these approaches, however, suffers from one or more serious disadvantages. For one thing, the wider the dissemination of the key, the more likely that its safety will be compromised. Similarly, the longer the key is in use—in terms of time alone or of the cumulative length of the transmissions sent—the more likely it is that it may be broken or discovered by an interloper. In addition, the greater the number of keys to be used—either for different recipients or for different time periods—the more difficult it is to assure that the proper key will be used for the particular transmission. Finally, the more often the parties need to obtain or forward the encoding keys, the more likely it is that the particular key will be intercepted during such information exchange, even if not only a different communication but also a different communications channel (such as a telephone) than that to be used for the coded data transmission (i.e. a data link) were to be employed to carry the information about the encoding key.

These and other deficiencies have lead to the development of additional alternatives to secure data transmission. One currently employed alternative approach, commonly referred to as RSA public key encryption, involves the use of a total of four encryption keys—two for each party, one public and the other private. Each party knows (e.g. is able to retrieve from a safe storage location) its own public and private keys, and is able to obtain the public key of the respective other party since that key, as its name implies, is available to the "public", or at least to the system users or subscribers. In use, two such keys are actually employed at each of the issuing and receiving ends. More particularly, the initiator or originator of the transmission (i.e. the party desiring to send an encrypted message) first encodes the message using his or her own private code key, and then re-encodes such encoded message using the other party's (intended recipient's) public code key. The thus doubly-encoded message is then sent to the intended recipient and must be decoded at that end before the original message can be deciphered. To this end, a double decoding process akin to the double encoding process is performed at the recipient end, first using the recipient's private key and then decoding the result by utilizing the sender's public key.

It will be appreciated that this approach is rather complex and cumbersome in that it requires double use of the respective coding (i.e. encoding or decoding, as the case may be) technique and/or equipment at each end, and cannot be performed (i.e. successfully commenced and concluded) unless each of the parties has the correct public key of the other party and uses it in conjunction with his or her own correct private key during the respective coding operation. The need for double coding and attendant entry of two different coding keys at each end of the transmission significantly increases the risk that machine or human error could result in the presentation of a garbled or otherwise indecipherable message to the intended recipient.

OBJECTS OF THE INVENTION

It is accordingly an object of the present invention to avoid the aforedescribed disadvantages of the prior art.

More particularly, it is an object of the invention to provide a method of securely transferring data between respective issuers and intended recipients, which method does not possess the disadvantages of previously proposed or utilized methods of this type.

Still another object of the present invention is to devise a method of the type here under consideration which avoids the need for prior knowledge at either of the transmission ends of any coding key being used at the respective other end.

A concomitant object of the invention is to develop a method of the above type that is relatively simple to implement and perform, and yet highly secure and reliable.

SUMMARY OF THE INVENTION

In keeping with these objects and others that will hereinafter become apparent, one feature of the present invention resides in a method for achieving secure data transmission between respective sending and receiving terminals of a telecommunication system. In accordance with one aspect of the present invention, this method comprises the steps of establishing a multiplicity of correlations each defining a relationship for pairing an arbitrary challenge data string in a unique and consistent manner with a different correspond-

ing response data string, and associating each of these multiple correlations with a corresponding one of the individual subscribers, including making each such correlation available to the corresponding individual subscriber and to the server device to enable the corresponding subscriber and the server device to generate one of the arbitrary challenge data string and the corresponding response data string from the other of the arbitrary challenge data string and the corresponding response data string using said each correlation. This method further includes apprising the server device of the identities of an originating subscriber and an intended receiving subscriber for a particular transmission. In accordance with the invention, a message to be conveyed in encrypted form in each particular transmission from the originating subscriber through the server device to the intended receiving subscriber is cryptographically processed. This processing includes the steps of providing a first arbitrary challenge data string to define an originating subscriber pair formed of the first arbitrary challenge data string and a first response data string generated from the first arbitrary challenge data string utilizing the correlation associated with the originating subscriber; generating, at the originating subscriber, the first response data string of the originating subscriber pair utilizing the first arbitrary challenge data string and the correlation associated with the originating subscriber; encrypting the message at the originating subscriber using one of the data strings of the originating subscriber pair; posting the encrypted message and the other of the data strings of the originating subscriber pair to the telecommunications system at the originating subscriber; receiving the encrypted message and the other of the data strings of the originating subscriber pair at the server device; generating the one of the data strings of the originating subscriber pair at the server device utilizing the correlation associated with the originating subscriber; and decrypting the encrypted message at the server device using the said one of the data strings of the originating subscriber pair to recover the message. According to the inventive method, there is further provided a second arbitrary challenge data string at the server device to define a receiving subscriber pair formed of the second arbitrary challenge data string and a second response data string generated from the second arbitrary challenge data string utilizing the correlation associated with the receiving subscriber. The processing further includes re-encrypting the recovered message at the server device using one of the data strings of the receiving subscriber pair, and posting the re-encrypted message and the other of the data strings of the receiving subscriber pair to the telecommunications system at the server device for delivery to the receiving subscriber. The re-encrypted message and the other of the data strings of the receiving subscriber pair are received at the receiving subscriber, the one of the data strings of the receiving subscriber pair is generated at the receiving subscriber utilizing the correlation associated with the receiving subscriber, and the re-encoded message is decrypted at the receiving subscriber using the one of the data strings of the receiving subscriber pair to recover the message from the originating subscriber.

A particularly advantageous implementation of the method of the present invention is obtained when each of the telecommunication system terminals, which are connected with one another through the intervening server device, is associated with an interface device operative for transferring data between the respective terminal and a respective system subscriber card. Moreover, in this preferred implementation, each of the subscriber cards includes at least a data storage and a processor for processing data obtained from the data storage and from the respective terminal and operable for issuing output data to the respective terminal. Each card is also individualized for the respective individual subscriber by storing in its data storage a code key data string that is unique to that subscriber. There is further provided a repository that is accessible to the server device and that stores or enables server access to at least an association between each individual subscriber and the code key data string stored on the individual subscriber's individualized subscriber card.

In that environment, this particular implementation of the method of the present invention is used to perform secure data transmission through the server device between the sending and receiving terminals, with some of the steps of the present method being performed at the sending terminal, others at the server device, and still others at the receiving terminal.

The steps taking place at the sending terminal include: providing a unique original sender challenge data string; transferring the original sender challenge data string to the respective individualized sending subscriber card; processing the original sender challenge data string and the unique code key data string on the respective sending subscriber card to obtain a sender response data string that has a first relationship to the original sender challenge data string, which relationship is unique to the respective sending subscriber card; encoding original data that is to be securely transmitted by one of the sender response and challenge data strings to provide encoded data; and transmitting the encoded data and the other of the sender challenge and response data strings, together with identification of the sending subscriber card in unencrypted form and further information identifying the intended recipient subscriber, to the server device.

These steps are followed by the following steps occurring at the server device: retrieving from the repository the code key data string associated with the thus-identified sending subscriber card; utilizing the unique first relationship determined by the thus-retrieved code key data string to obtain the one from the other of the sender challenge and response data strings; decoding the encoded data utilizing the thus-obtained one of the sender response and challenge data strings; retrieving from the repository the unique recipient subscriber code key data string associated with the subscriber card issued to the intended recipient subscriber as identified in the further information; providing a unique original server challenge data string; processing the original server challenge data string and the retrieved unique recipient subscriber code key data string in the same manner as they would be on the intended recipient subscriber card to obtain a server response data string that has a second relationship to the original server challenge data string, which second relationship is tailored for the respective intended recipient subscriber card; re-encoding the previously decoded data by one of the server response and challenge data strings to provide re-encoded data; and transmitting the re-encoded data, together with the other of the server challenge and response data strings, to a respective receiving terminal associated with the recipient subscriber.

The method is then completed by performing the following steps at the receiving terminal: transferring the other of the server challenge and response data strings to the respective individualized recipient subscriber card; processing the thus-transferred other of the server challenge and response data strings and the unique code key data string on the respective recipient subscriber card to obtain a recipient response data string that corresponds to the one of the server

challenge and response data strings when conducted on the intended recipient subscriber card in accordance with the second relationship; and decoding the re-encoded data utilizing the thus-obtained one of the server response and challenge data strings to provide an unencrypted replica of the original data.

It will be appreciated that the method of the present invention as heretofore described greatly simplifies the message encoding/decoding process for both the issuer and the intended recipient of the message in that it is accomplished with neither the issuer nor the intended recipient having to use the respective other subscriber's coding key. As a matter of fact, the respective subscriber does not even have to know or have direct access to the other subscriber's coding key or, for that matter, his or her own such key. Moreover, the process can proceed without either subscriber having to know either the challenge or the response applied at the other subscriber's end or, provided that the respective terminal is programmed to generate the response and perform the coding operation without outside input, even those applicable at his or her own end. In other words, the entire coding process may be implemented so as to be transparent to the two subscribers, e.g. by using a computer-generated random number as the challenge applied at the issuing or sending terminal, thus relieving the communicating subscribers of the burden of obtaining and entering any codes or other data strings to be used in the coding process at his or her end. Furthermore, so long as the code key is not revealed, there is no need to take special precautions beyond those needed to maintain the confidentiality of the message itself in order to conceal the respective challenge or response data string applied at the respective subscriber's end, assuming that such information is available there to begin with.

It is particularly advantageous when, in accordance with an aspect of the present invention, at least one of the encoding and re-encoding steps includes employing the respective response data string to provide the respective encoded or re-encoded data; and wherein that of the transmitting steps which comes up first after the said one of the encoding and re-encoding steps includes sending the respective encoded or re-encoded data accompanied by the respective challenge in unencrypted form. In this further scenario, it is also advantageous when that of the processing and utilizing steps which occurs just prior to the said one of the encoding and re-encoding steps includes utilizing a predetermined algorithm to form the respective response data string in response to the respective challenge data string; and wherein that of the utilizing and processing steps which comes after the aforementioned one of the encoding and re-encoding steps includes utilizing the same predetermined algorithm to make the respective response data string formed in response to the respective challenge data string identical to that used in the one of the encoding and re-encoding steps for use in the following one of the decoding steps.

An important advantage of this approach is that the relationship between the challenge and response data strings need not be symmetrical or reversible in the sense that there need exist an inverse relationship or algorithm operative for unambiguously reconstituting the original challenge data string from the response dam string and the respective subscriber code key, and yet the coding process can proceed. This is so because the same conversion process or algorithm is used at both of the affected stations (i.e. the sending terminal and the decoding part of the server device, or the re-encoding part of the server device and the receiving terminal) in the same direction (i.e. from the unencrypted challenge data string to the response data string), so that a need to proceed in the opposite direction does not arise.

The novel features which are considered as characteristic of the invention are set forth in particular in the appended claims. The improved method of performing secure data transmission itself, together with additional features and advantages thereof, will however be best understood upon perusal of the following derailed description of certain specific embodiments with reference to the accompanying drawing.

BRIEF DESCRIPTION OF THE DRAWING

In the drawings, wherein like reference numerals identify similar elements throughout the several views:

FIG. **1** is a block diagram showing pertinent portions of a transmission system apparatus that may be employed in the practice of the method of the present invention; and

FIG. **2** is another block diagram illustrating various electronic components and circuitry provided on a typical individualized subscriber card that may be used in the system of FIG. **1** for performing encoding of messages and the like in accordance with the method of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring now to the drawings, and initially to FIG. **1**, it may be seen that the reference numeral **10** has been used therein to identify a telecommunications system of a type suited for performing encryption and decryption processes in accordance with the present invention. The system **10** is shown diagrammatically, and only to the extent necessary to describe and enable a clear understanding of the present invention.

As is well known, the telecommunication system **10** typically includes a very large number of subscriber terminals or stations **11.1** to **11.n** (with n denoting any positive integer number exceeding one); however, only those two terminals specifically mentioned above and various devices and/or items associated with each have been illustrated since this is all that is needed to fully describe the inventive method. To further simplify the following explanation, each of the terminals **11.1** to **11.n**, together with all associated devices and items, will be collectively referred to as terminal equipment **12.1** to **12.n**, and only the terminal equipment **12.1** will be described in some detail as to its basic construction and operation. However, it should be understood that each additional terminal equipment **12.n** is identical to the terminal equipment **12.1**, if not in all details then at least in those respects, including the presence of corresponding components, that make the various units of terminal equipment **12.1** to **12.n** compatible with one another in the sense of being capable of exchanging and processing a variety of data from various sources.

In a currently preferred implementation of the present invention, the terminal equipment **12.1** includes, besides the terminal **11.1** itself, a card reader **13.1** that is constructed, in a known manner, to at least read or retrieve data stored on a compatible subscriber card **14.1**. While the card **14.1** is portable and may be used at any of the terminals **11.1** to **11.n**, it will for the sake of simplicity be treated here as an item associated with the terminal **11.1** alone and thus constituting a component of the terminal equipment **12.1**. For the card **14.1** to be fully capable of use in accordance with all aspects of the present invention, it is contemplated that it advantageously be of at least of the so-called "smart card", if not the

"super-smart card", type, as will be explained below in conjunction with FIG. 2. Of course, the card reader 13.1 should be compatible with the card 14.1, whatever its type or capabilities or features.

The card reader 13.1 is connected, as by a data line or bus 15.1, with an input/output (I/O)unit 16.1 of the terminal 11.1. The I/O unit 16.1 is connected with a central processing unit (CPU) 17.1 that, in turn, is connected with a memory or data storage unit 18.1. For purposes of discussion, it will be assumed that the memory unit 18.1 contains the text of a message that is to be sent to the terminal 11.n after it has been encrypted for security during transmission. In this scenario, it will also be assumed that the memory unit 18.n is intended to store the text of the message after it has been decrypted. The actual text encryption and decryption may and typically will be performed by the respective one of the CPU units 17.1 to 17.n. It should be apparent that data transmission can take place from any one of the terminals 11.1 to 11.n to any other, including in a direction opposite to that assumed here, i.e. from the terminal 11.n to the terminal 11.1. Nevertheless, the operation of the system 10 in accordance with the method of the present invention will by way of example be explained here as applied to a situation in which the terminal 11.1 is the originating or transmitting, and terminal 11.n is the receiving, station or terminal.

The transmission between the terminals 11.1 and 11.n in the system 10 of the present invention does not take place directly from terminal to terminal; rather, the I/O units 16.1 of all of the terminals 11.1 to 11.n are connected or linked, through respective data transmission channels (e.g. shielded data transmission lines, cables, busses, optical fiber cables or wireless links or the like) 19.1 to 19.n, with a server device 20. The server device 20 itself, as well as the information contained therein, is maintained in a highly protected, secure manner by the owner or operator of a telecommunication network incorporating or used in conjunction with the system 10. The server device 20 is connected for communication with, or may even physically incorporate, a repository 21 of certain information, including that associating individual subscriber cards 14.1 to 14.n that have been personalized or individualized prior to their issuance to individual subscribers to the system 10, with such subscribers and with secret key codes that have been assigned to each such subscriber in the course of the individualization process. This association information is maintained in the repository 21 in a secure and highly confidential manner. Moreover, while the server device 20 is able to access and retrieve the confidential association and key number information from the repository 21, it can use it for only limited purposes, and well known or otherwise appropriate measures are taken to assure that this information not be released to unauthorized persons, including server device operating personnel with no need to know, or to unauthorized destinations (e.g. computer hacker terminal equipment), or for unauthorized uses.

The safeguarding of this information, which is highly sensitive because its release could result in compromising the secrecy of encrypted communications issued or received by a subscriber affected by the revealing of this information, goes even beyond the securement of the server device 20 in that, as currently preferred, even the subscriber himself or herself does not know or have direct access to his or her code key, although the same is stored by the subscriber's personalized subscriber card 14.1. To this end, the card 14.1 (like each other such card 14.n) may include, as indicated in FIG. 2 of the drawings, a memory 22.1 having at least a portion 23.1 that is protected in the sense that the data contained therein can only be accessed for limited purposes and cannot be "data-dumped", i.e. extracted from the respective card 14.1 either directly or through some manipulation of either the data or of the card. Suitable measures used to achieve this safeguarding feature are known to those skilled in this art and need not be elaborated upon here, particularly since they do not form a part of the present invention.

As also shown in FIG. 2, the respective subscriber card 14.1 further includes a processor 24.1 capable of retrieving information contained in the memory 22.1, inclusive of that secretly maintained in the memory portion 23.1, and of processing such information together with other information or data, as for example that obtained from an input/output (I/O) device 25.1 that is also present on the card 14.1, to generate output information and issue the same through the I/O device 25.1 to the card reader 13.1 for further handling thereat and/or transmission to the terminal 11.1, all in a manner hereinafter described.

Inasmuch as the card 14.1 includes at least basic data storage and processing devices and circuitry, it qualifies for the designation "smart card" that is increasingly being used to describe passive cards with data processing capability—as distinguished from the traditional magnetic strip and similar "memory only" cards. The card 14.1 of this type is not equipped to permit the user or subscriber enter any data into it, and ordinarily does not even include means for displaying any data available in or from the card 14.1. Such capabilities, if needed, must accordingly be provided by other equipment such as the card reader 13.1, the terminal 11.1 and/or keyboard and/or display equipment associated therewith (not shown). In any event, as indicated in broken lines in FIG. 2, the card 14.1 may alternatively be of the "super-smart" type, being in such case additionally provided with a mini-keyboard 26.1 connected with the processor 24.1 to supply data entries thereto, and/or a display device (such as a liquid crystal display) 27.1 for receiving data to be displayed from the processor 24.1 through the I/O device 25.1 and for presenting that data in a visually perceptible form.

Cards of the type thus described are currently available from several sources and under a variety of appellations, including as AT&T Smart Cards, as are the associated card readers and other terminal equipment so that it is not necessary, at this point to describe them in further detail. Suffice it to say that the equipment and circuitry described above, albeit largely conventional in nature, is configured in such manner as to perform certain hereinafter-described tasks in accordance with the present invention.

Having thus described the structural elements and arrangements of the system 10, the features of the currently preferred implementation of the present invention will now be discussed as employed in or in conjunction with the system 10 for generating an encrypted message at the sending or originating terminal 11.1 and for decrypting it at the receiving terminal 11.n, using the respective subscriber cards 14.1 and 14.n at the respective transmission link ends or terminals 11.1 and 11.n.

The operation of the system 10 in accordance with this implementation of the invention commences at the origination or issuing terminal 11.1 in that, when a message is to be sent in a secured (encoded or encrypted) fashion, a challenge data string is presented to the processor 24.1 of the card 14.1. This challenge data string may be internally generated either at the terminal 11.1 or on the card 14.1, or even by the server device 20, as by a random number generator; however, if preferred, the challenge data string may instead by arbi-

trarily selected and manually or otherwise entered by the subscriber, as for example through the mini-keyboard **26.1** if provided, or through a keyboard associated with the terminal **11.1**, or through a similar or functionally equivalent device incorporated in or associated with the card reader **13.1** (the latter two possibilities not having been illustrated). On receipt of the challenge data string, the processor **24.1** located on the card **14.1** retrieves the subscriber's own secret code key from the memory portion or secret memory cache **23.1**, and processes the two data strings in accordance with a predetermined or known protocol or algorithm to obtain a response data string. At this juncture, it should be mentioned that even though the algorithm used to arrive at the response data string from the challenge data string is contemplated as being the same for all of the subscriber cards **14.1** to **14.n** (and the same as that used at the server device **20**, as discussed below), the response data string generated at each of the subscriber cards **14.1** to **14.n** will be unique to that subscriber card—that is, different from the response data string obtained from any other of the cards **14.1** to **14.n** if any such other card were presented with the identical challenge data string. This different card "reaction" to the challenge string results from the use of the different, unique subscriber code key stored in or on each of the cards **14.1** to **14.n**. It will nevertheless be appreciated that the foregoing approach, while currently preferred, is not the only one that may be used in accordance with the present invention to obtain the desired unique relationship or correlation between any arbitrarily chosen challenge data string and the corresponding response data string associated with the particular subscriber and/or his or her card **14.1** to **14.n**. Another implementation contemplates a look-up or conversion table instead of or in combination with an appropriate conversion or processing algorithm. In any event, what is important in the context of the present invention is that the correlation applicable to each pair of challenge and response data strings, while different from one subscriber to another, be consistent or reproducible insofar as each particular subscriber is concerned—i.e. that the response data string corresponding to a particular challenge data string is always the same, no matter how many attempts are made and at what location, so long as the correlation associated with the particular subscriber is used to arrive at one of the members of such data string pair from the other.

With the unique card response data string having been generated on the card **14.1**, the actual encoding of the message to be transmitted can commence, in one of two presently-contemplated manners using the challenge and response data strings. One is to employ the challenge data string as the encryption key for the message, in which case the response data string would then be sent with the challenge-encrypted message to the server device **20** for use in further handling (i.e. decryption) of the message. The other is to encrypt the message using the response data string and to then send the original challenge data string with the response-encrypted message for similar use at the server device **20**. It is currently preferred to use the latter approach in the aforementioned implementation, and the following description will assume that approach; however, it will be appreciated that the principles described in conjunction with this particular method are equally applicable, with only a minimum of self-evident adaptation, to the other alternative as well. The exact algorithm used to encrypt the data or message using either the challenge or the response data string is substantially a matter of design choice.

When the encrypted message and the accompanying information sent with it—which may include, in addition to the challenge data string, at least some information to enable the server device **20** to determine from whom or where the transmission originated (i.e. the identity of the owner of the subscriber card **14.1**)—is received by the server device **20**, the latter retrieves the code key that is associated with the originator's subscriber card **14.1** (i.e. the code key that is stored on the card **14.1**) from the repository **21**. The server device **20** then processes the thus-retrieved subscriber code key and the (unencrypted) original challenge data string that had been received from the terminal **11.1** as a part of the transmission, in the same way (i.e. using the same algorithm or correlation) as that used by the originating subscriber to generate the response data string with which the original message was encrypted. Using the same starting values and processing operations at the server device **20** as were used at the subscriber site, the server device **20** thus generates a response data string to the challenge data string that is identical to that which was generated using the subscriber card **14.1** and used to encode the original message received by the server device **20**. The server device **20** then uses this information to decrypt the encoded message and to thereby restore it to its original unencrypted state.

As will be apparent, since the server device **20** is only an intermediary in the transmission and not the intended ultimate recipient, steps must be taken to safeguard the message as it is next transmitted from the server **20** to the recipient terminal **11.n**. Toward that end, the message is re-encrypted before it is transmitted from the server device **20** to the receiving terminal **11.n**; in brief, the server device **20** employs substantially the same technique as that described above between the originating subscriber and the server device **20**, but now geared toward the ultimate recipient. First, the server device **20** determines, as from the message itself—where this information may be a part of the encoded text (cryptotext), or may have accompanied the encrypted text in unencrypted (plaintext) form—the identity of the intended ultimate recipient and correspondingly, identifies the subscriber card **14.n** issued to that subscriber. With this information, the server device **20** accesses the repository **21** to retrieve the secret code key for that subscriber and which is embedded in the restricted memory portion **23.n** of the receiving subscriber card **14.n**. From there, the process is the same as that discussed above—the server device **20** produces a response data string to a locally-generated random number challenge data string using the same algorithm as before, uses the resulting response data string to re-encrypt the previously deciphered or decrypted message, and sends the so re-encoded message, together with new unencrypted challenge data string, to the recipient terminal **11.n**.

Upon receipt of this re-transmission at the recipient terminal **11.n**, or as and when instructed to do so at some later time by the recipient subscriber, the challenge data string is sent to the card reader **13.n**. Assuming that the proper card **14.n** (i.e. that of the intended recipient) is present in the card reader **13.n**, the challenge data string is processed by the processor **24.n** of the card **14.n** using the same code key (this time retrieved from the memory **22.n** of the subscriber card **14.n**) and the same algorithm or correlation as that previously used by the server device **20** to produce the identical response data string with which the message was re-encrypted. The resulting response data string generated by the processor **24.n** of the subscriber card **14.n** is then used by the receiving subscriber at the terminal **11.n** to decrypt the server re-encrypted message—i.e. to restore it to the original, unencrypted form with which the originating subscriber initiated the transmission process.

An example of a transmission in accordance with the

invention will now be described to further illustrate the features explained above, using italics to indicate unencrypted portions of the transmission, square brackets to indicate encrypted or re-encrypted portions of the transmission, quotation marks for identification information, and parentheses for explanatory matter.

The transmission sent from the originating terminal **11.1** may by way of example read as follows:

(header:) to "server" from "subscriber1"

(body:) challenge1,[message1],

wherein [message1]contains:

[(header:) to "subscriber2"

(body:) message],

and wherein response1 (to challenge1) has been used to encode the [message1].

Using the thus-received "subscriber1" information, the server device **20** retrieves the associated code key from the repository **22** and uses it, together with challenge1, to recreate response1. It then uses the thus-obtained response1 to regenerate (decrypt) message1 from [message1]. That will reveal the "subscriber2" information to the server device **20** and permit the "subscriber2" information to be used to similarly retrieve the intended recipient code key from the repository **21** and to use the retrieved recipient code key, together with a locally-generated challenge2, to produce a response2 that is then used to form a re-encrypted [message1] which is included in a transmission sent to the terminal **14**.n associated with "subscriber2" and reading, basically, as follows:

(header:) to "subscriber2" from "server"

(body:) challenge2, [message2],

wherein [message2] contains:

[(header:) from "subscriber1"

(body:) message].

The intended recipient, after becoming aware of the arrival of such transmission, uses his or her subscriber card **14**.n to generate the response2 to the server-transmitted challenge2, and to decrypt [message2] using the thus obtained response2 to restore message2 and thus message to its original, usually plaintext form.

The example presented above has particular but not exclusive utility when it is desired to conceal the fact that the originating and receiving subscribers are communicating. Incipient business merger contacts and negotiations between warring political entities may provide illustrative examples of situations in which the release of information that particular parties are "talking" could be detrimental, irrespective of the actual contents of the exchanged messages. However, where for example this information is not sensitive, the identification of the intended recipient may be indicated by the originating subscriber directly or outside of the "envelope" containing (i.e. as a plaintext part of the header of) the encrypted [message1]. In other words, the originating header might then read: to "subscriber2" from "subscriber1", and the same or similar language might also be used in the header sent by the server device **20** to the intended recipient. The transmission would nonetheless still be intercepted by the server device **20** between the originating and receiving subscribers and processed thereat in a manner substantially identical to that described above. The involvement of or intervention by the server device **20** in the transmission would, in such an arrangement, be transparent to both the originator and the recipient of the transmission.

While the invention has been illustrated and described as embodied in a particular arrangement and apparatus, it is not intended to be limited to the details shown since various modifications and structural changes may be made without departing in any way from the spirit of the present invention.

Without further analysis, the foregoing will so fully reveal the gist of the present invention that others can, by applying current knowledge, readily adapt it for various applications without omitting features that, from the standpoint of prior art, fairly constitute essential characteristics of the generic and specific aspects of the contribution to the art and, therefore, such adaptations should and are intended to be comprehended as within the meaning and range of equivalence of the claims. What is claimed as new and desired to be protected by Letters Patent is set forth in the appended claims.

What is claimed is:

1. A method of performing secure data transmission of messages between individual subscribers of a telecommunications system that includes individual terminals linked by a server device, comprising the steps of:

establishing a multiplicity of correlations each defining a relationship for pairing an arbitrary challenge data string in a unique and consistent manner with a different corresponding response data string;

associating each of said multiple correlations with a corresponding one of the individual subscribers, and making said each correlation available to the corresponding individual subscriber and to the server device to enable the corresponding subscriber and the server device to generate one of the arbitrary challenge data string and the corresponding response data string from the other of the arbitrary challenge data string and the corresponding response data string using said each correlation;

apprising the server device of the identities of an originating subscriber and an intended receiving subscriber for a particular transmission; and

cryptographically processing a message to be conveyed in encrypted form in each said particular transmission from the originating subscriber through the server device to the intended receiving subscriber, comprising the steps of:

providing a first arbitrary challenge data string to define an originating subscriber pair formed of the first arbitrary challenge data string and a first response data string generated from the first arbitrary challenge data string utilizing the correlation associated with the originating subscriber;

generating, at the originating subscriber, the first response data string of the originating subscriber pair utilizing the first arbitrary challenge data string and the correlation associated with the originating subscriber;

encrypting the message at the originating subscriber using one of the data strings of the originating subscriber pair, and posting the encrypted message and the other of the data strings of the originating subscriber pair to the telecommunications system at the originating subscriber;

receiving the encrypted message and the other of the data strings of the originating subscriber pair at the server device, generating the one of the data strings of the originating subscriber pair at the server device utilizing the correlation associated with the originating subscriber, and decrypting the encrypted message at the server device using the said one of the data strings of the originating subscriber pair to recover the message;

providing a second arbitrary challenge data string at the

server device to define a receiving subscriber pair formed of the second arbitrary challenge data string and a second response data string generated from the second arbitrary challenge data string utilizing the correlation associated with the receiving subscriber;

re-encrypting the recovered message at the server device using one of the data strings of the receiving subscriber pair, and posting the re-encrypted message and the other of the data strings of the receiving subscriber pair to the telecommunications system at the server device for delivery to the receiving subscriber;

receiving the re-encrypted message and the other of the data strings of the receiving subscriber pair at the receiving subscriber, generating the one of the data strings of the receiving subscriber pair at the receiving subscriber utilizing the correlation associated with the receiving subscriber, and decrypting the re-encoded message at the receiving subscriber using the said one of the data strings of the receiving subscriber pair to recover the message from the originating subscriber.

2. The method as defined in claim 1, wherein said associating step includes providing each said individual subscriber with an individualized card that includes a data storage containing an individual code key unique to said corresponding individual subscriber, storing said individual code keys for all of said individual subscribers in a repository accessible to the server device in a manner identifying each said key code with the corresponding individual subscriber, and providing each of said individual subscribers and said server device with access to an algorithm that establishes the correlation of said each individual subscriber when used in conjunction with the code key of said each individual subscriber, and wherein said step of making said each correlation available includes retrieving the respective code key and said algorithm at each of said individual subscribers and at said server device for use in said generating steps.

3. The method as defined in claim 2, wherein said associating step further includes providing each of said individual subscriber cards with data processing capability; and wherein said steps of generating said first response data string of the originating subscriber pair at the originating subscriber and said one of said data strings of the receiving subscriber pair at the receiving subscriber include processing said first arbitrary challenge data string and said other of said receiving subscriber data strings on a respective one of the originating and receiving subscriber cards utilizing the correlation associated with the originating and with the recciving subscriber, respectively, as obtained from the respective subscriber card.

4. A method of performing secure data transmission between respective sending and receiving telecommunication system terminals that are connected with one another through a server device, each of said terminals being associated with an interface device operative for transferring data between the respective terminal and a respective system subscriber card that includes at least a data storage and a processor for processing data obtained from the data storage and from the respective terminal and operable for issuing output data to the respective terminal, each of the subscriber cards being individualized prior to issuance thereof to a respective individual subscriber by storing in its data storage a unique code key data string, with at least an association between each individual subscriber and the unique code key data string stored on that individual subscriber's individualized subscriber card being stored in a repository accessible to the server device, comprising the steps of:

(A) at a respective sending terminal:

(i) providing a unique original sender challenge data string;

(ii) transferring the original sender challenge data string to the respective individualized sender subscriber card;

(iii) processing the original sender challenge data string and the unique code key data string on the respective sender subscriber card to obtain a sender response data string that has a first relationship to the original sender challenge data string, which relationship is unique to the respective sending subscriber card;

(iv) encrypting original data that is to be securely transmitted by the sending subscriber using one of the sender response data string and the challenge data string to provide encrypted data; and

(v) transmitting the encrypted data and the other of the sender challenge data string and the response data string to the server device;

(B) at the server device:

(i) receiving from the sending terminal the encrypted data and the other of the sender challenge data string and the response data string;

(ii) retrieving from the repository the code key data string associated with the subscriber card of the sending subscriber;

(iii) utilizing the unique first relationship of the retrieved code key data string to obtain the one from the other of the sender challenge data string and the response data string;

(iv) decrypting the received encrypted data utilizing the thus obtained one of the sender response data string and the challenge data string;

(v) retrieving from the repository the unique recipient subscriber code key data string associated with the subscriber card issued to the intended recipient subscriber;

(vi) providing a unique server challenge data string;

(vii) processing the server challenge data string and the retrieved unique recipient subscriber code key data string to obtain a server response data string that has a second relationship to the original server challenge data string, which relationship is tailored for the respective intended recipient subscriber card;

(viii) re-encrypting the server-decrypted data using one of the server response data string and the server challenge data string to provide server re-encrypted data; and

(ix) transmitting the server re-encrypted data, together with the other of the server challenge data string and the server response data string, to a respective receiving terminal associated with the recipient subscriber; and

(C) at the respective receiving terminal:

(i) receiving from the server device the server re-encrypted data and the other of the server challenge data string and the server response data string;

(ii) transferring the other of the server challenge data string and the server response data string to the respective individualized recipient subscriber card;

(iii) processing the received other of the server challenge data string and the server response data string and the unique code key data string on the respective recipient subscriber card to obtain a resultant recipient data string corresponding to the one of the server

challenge data string and the server response data string used for the encryption at the server device; and

(iv) decrypting the server re-encrypted data utilizing the one of the received server response data string and the server challenge data string to provide a replica of the original data.

5. The method as defined in claim 4, wherein at least one of said encrypting and re-encrypting steps includes employing the respective response data string to provide one of the respective encrypted and re-encrypted data; and wherein said step of transmitting said one of said encrypted and re-encrypted data includes sending the respective challenge data string with such data.

6. The method as defined in claim 5, wherein at least one of said processing steps and an associated one of said utilizing steps use an identical algorithm to form the respective response data string in response to the respective challenge data string.

7. A method of performing secure data transmission of messages between individual subscribers of a telecommunications system that includes individual terminals linked by a server device, comprising the steps of:

establishing a multiplicity of correlations each defining a relationship for pairing an arbitrary challenge data string in a unique and consistent manner with a different corresponding response data string;

making each of said multiple correlations available to a corresponding one of the individual subscribers, and all of said multiple correlations available to the server device in identifying associations with said corresponding individual subscribers;

apprising the server device of the identities of an originating subscriber and an intended receiving subscriber for a particular transmission;

cryptographically processing an original message to be conveyed in encrypted form in said particular transmission from the originating subscriber through the server device to the intended receiving subscriber, comprising the steps of:

generating an originating subscriber pair of the challenge and response data strings at one of said originating subscriber and server device utilizing the correlation associated with the originating subscriber;

communicating one of the data strings of said originating subscriber pair from said one to the other of said originating subscriber and server device;

regenerating the other of the data strings of said originating subscriber pair at the other of said originating subscriber and server device from the thus communicated one data string;

encrypting the message at the originating subscriber using said other of the data strings of the originating subscriber pair;

posting the encrypted message to the telecommunications system at the originating subscriber;

receiving the encrypted message at the server device;

decrypting the encrypted message at the server device using the said other of the data strings of the originating subscriber pair to recover the message; and

cryptographically reprocessing the thus recovered message at the server device and at the receiving subscriber in a manner corresponding to that employed in the performance of said cryptographically processing step at the originating subscriber and at the server, utilizing the correlation associated with the receiving subscriber in place of that associated with the originating subscriber.

* * * * *

US005231668A

United States Patent [19]

Kravitz

[11] Patent Number: 5,231,668

[45] Date of Patent: Jul. 27, 1993

[54] **DIGITAL SIGNATURE ALGORITHM**

[75] Inventor: **David W. Kravitz**, Owings Mills, Md.

[73] Assignee: **The United States of America, as represented by the Secretary of Commerce**, Washington, D.C.

[21] Appl. No.: **736,451**

[22] Filed: **Jul. 26, 1991**

[51] **Int. Cl.**[5] **H04K 1/00**

[52] **U.S. Cl.** **380/28**; 380/30

[58] **Field of Search** 380/28, 30

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,200,770	4/1980	Hellman	380/30
4,218,582	8/1980	Hellman	380/30
4,405,829	9/1983	Rivest	380/30
4,424,414	1/1984	Hellman	380/30
4,641,346	2/1987	Clark	380/51
4,748,668	5/1988	Shamir et al.	380/30
4,881,264	11/1989	Merkle	380/28
4,933,970	6/1990	Shamir	380/30
4,995,082	2/1991	Schnorr	380/30
5,005,200	4/1991	Fischer	380/30
5,097,504	3/1992	Camion et al.	380/30

OTHER PUBLICATIONS

C. P. Schnorr, letter (8 pages) to Director, Computer Systems Laboratories, Attn: Proposed FIPS, Oct. 30, 1991.

El Gamal, Taher, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IIII Transactions and Information Theory, vol. IT-31, No. 4, Jul. 1985.

Primary Examiner—Salvatore Cangialosi

Attorney, Agent, or Firm—Schnader, Harrison, Segal & Lewis

[57] **ABSTRACT**

A method is provided for generating and verifying a digital signature of a message m. This method requires a pair of corresponding public and secret keys (y and x) for each signer, as well as a pair of public and secret values (r and k) generated for each message by the signer. The public value r is calculated according to the rule $r=(g^k \bmod p) \bmod q$. A value s is then selected according to the rule $s=k^{-1}(H(m)+xr) \bmod q$ where H is a known conventional hashing function. The message m, along with the signature (r,s) is then transmitted. When the transmitted signal is received a verification process is provided. The received values of r and s are tested to determine whether they are congruent to 0 mod g. Additionally, r is tested to determine whether it is equal to v mod q, where v is computed from r, s, m and y. For legitimately executed signatures, $v=g^k \bmod p$.

44 Claims, 3 Drawing Sheets

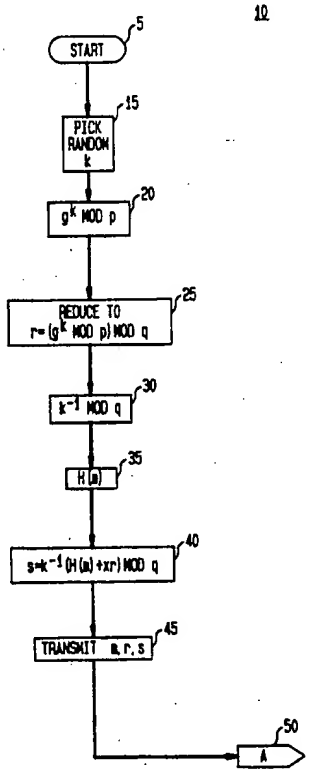


FIG. 1

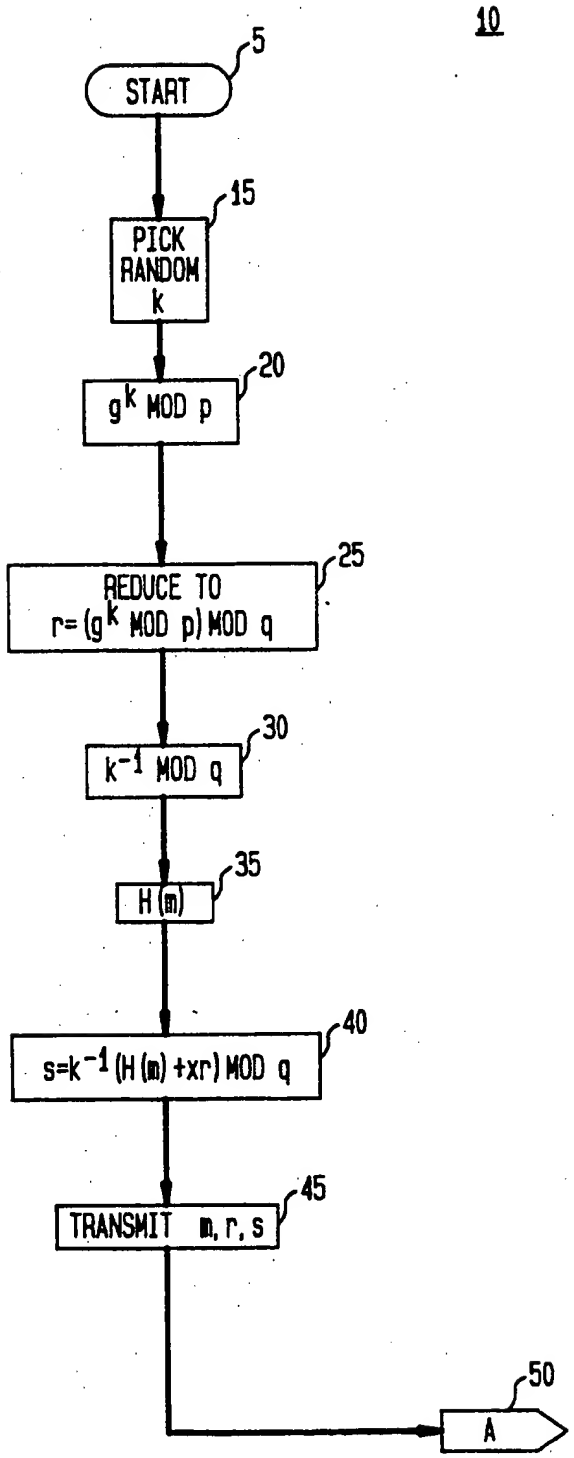

10
5
START
15
PICK RANDOM k
20
g^k MOD p
25
REDUCE TO $r=(g^k \text{ MOD } p) \text{ MOD } q$
30
k^{-1} MOD q
35
H(m)
40
$s=k^{-1}(H(m)+xr) \text{ MOD } q$
45
TRANSMIT m, r, s
50
A

FIG. 2

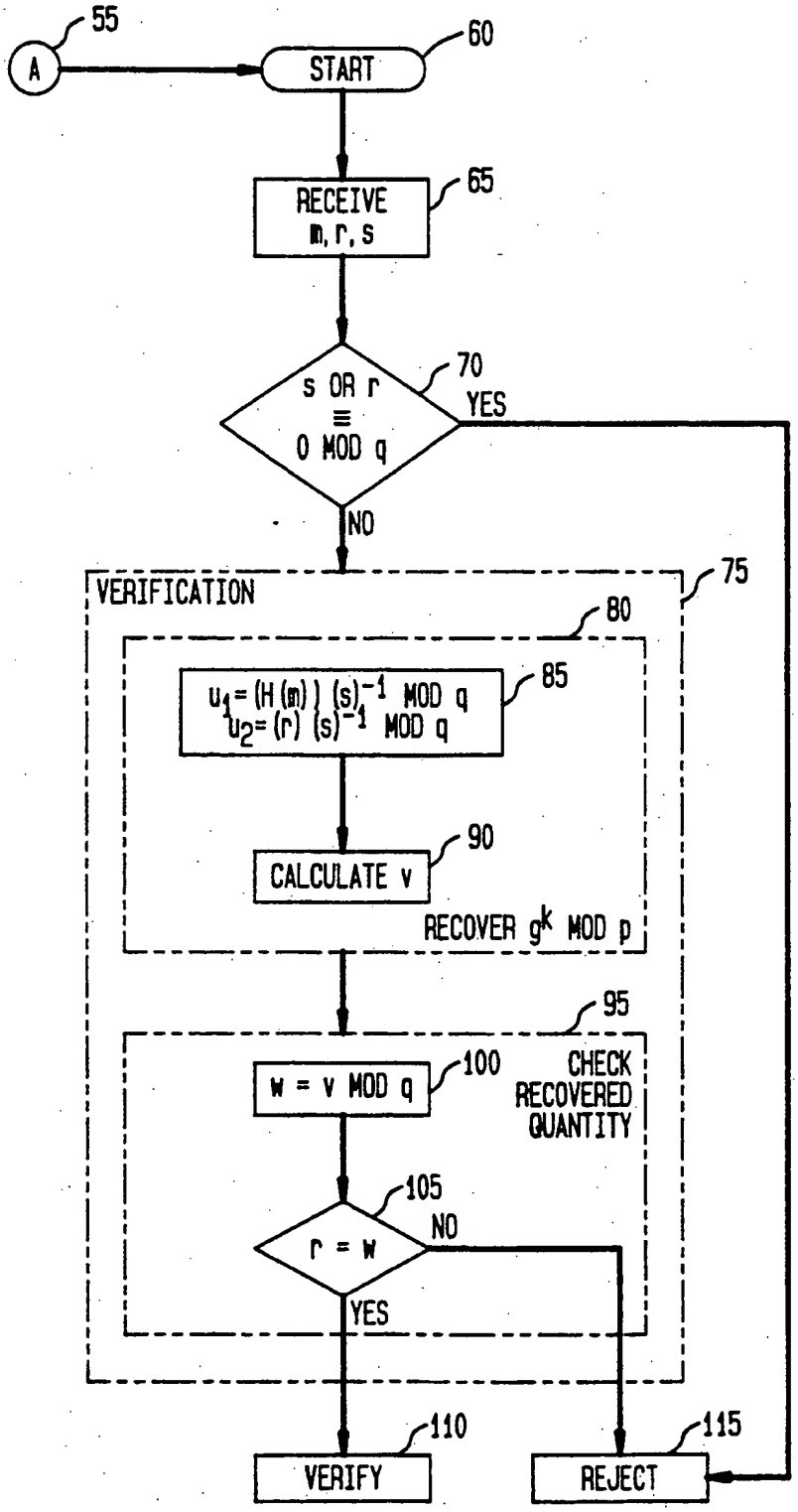
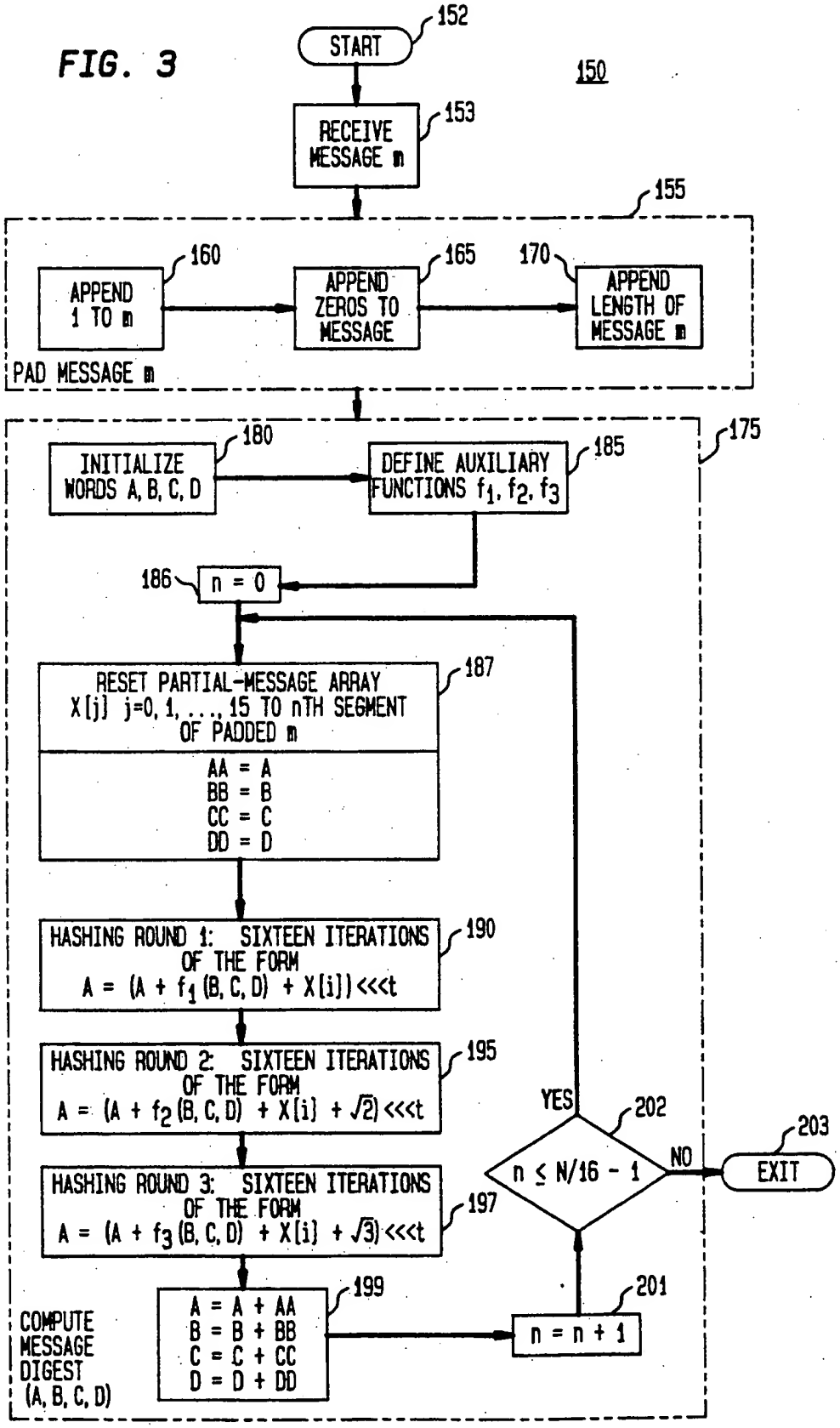

55
A
START
60
RECEIVE
m, r, s
65
s OR r
≡
0 MOD q
70
YES
NO
VERIFICATION
75
80
$u_1 = (H(m))(s)^{-1}$ MOD q
$u_2 = (r)(s)^{-1}$ MOD q
85
CALCULATE v
90
RECOVER g^k MOD p
95
w = v MOD q
100
CHECK RECOVERED QUANTITY
105
r = w
NO
YES
VERIFY
110
REJECT
115

FIG. 3


150
START
152
RECEIVE MESSAGE m
153
155
APPEND 1 TO m
160
APPEND ZEROS TO MESSAGE
165
APPEND LENGTH OF MESSAGE m
170
PAD MESSAGE m
175
INITIALIZE WORDS A, B, C, D
180
DEFINE AUXILIARY FUNCTIONS f_1, f_2, f_3
185
186
n = 0
RESET PARTIAL-MESSAGE ARRAY X[j] j=0, 1, . . ., 15 TO nTH SEGMENT OF PADDED m
187
AA = A
BB = B
CC = C
DD = D
HASHING ROUND 1: SIXTEEN ITERATIONS OF THE FORM
$A = (A + f_1(B, C, D) + X[i]) <<< t$
190
HASHING ROUND 2: SIXTEEN ITERATIONS OF THE FORM
$A = (A + f_2(B, C, D) + X[i] + \sqrt{2}) <<< t$
195
HASHING ROUND 3: SIXTEEN ITERATIONS OF THE FORM
$A = (A + f_3(B, C, D) + X[i] + \sqrt{3}) <<< t$
197
A = A + AA
B = B + BB
C = C + CC
D = D + DD
199
COMPUTE MESSAGE DIGEST (A, B, C, D)
YES
202
$n \leq N/16 - 1$
NO
203
EXIT
n = n + 1
201

DIGITAL SIGNATURE ALGORITHM

BACKGROUND OF THE INVENTION

1) Field of the Invention

The field of this invention is data integrity, and in particular generating and verifying a digital signature for a message or data file.

2) Background Art

When a message is transmitted from one party to another, the receiving party may desire to determine whether the message has been altered in transit. Furthermore, the receiving party may wish to be certain of the origin of the message. It is known in the prior art to provide both of these functions using digital signature algorithms. Several known digital signature algorithms are available for verifying the integrity of a message. These known digital signature algorithms may also be used to prove to a third party that the message was signed by the actual originator.

The use of public key cryptography to achieve instantiations of these digital signature algorithms is also known in the art. For example, Diffie and Hellman teach using public key cryptography to derive a digital signature algorithm in "New Directions in Cryptography," IEEE Transactions on Information Theory, Vol. IT-22 pp. 472–492, 1976. See also U.S. Pat. No. 4,200,770. Since then, several attempts have been made to find practical public key signature techniques which depend on the difficulty of solving certain mathematical problems to make message alteration or forgery by unauthorized parties difficult. For example, the Rivest-Shamir-Adleman system depends on the difficulty of factoring large integers. See R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Communications of the ACM, Feb. 1978, Vol. 21, No. 2, pp. 120–126, and U.S. Pat. No. 4,405,829.

Taher ElGamal teaches a signature scheme in "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms" in IEEE Transactions on Information Theory, Vol. IT-31, No. Jul. 4, 1985. It is believed that this system relies on the difficulty of computing discrete logarithms over finite fields. In the system taught by ElGamal m denotes a document to be signed, where $0 \leqq m \leqq p-2$ where p is a large prime and α is a primitive element mod p, both known. In any of the cryptographic systems based on discrete logarithms, p must be chosen such that $p-1$ has at least one large prime factor. If $p-1$ has only small prime factors, then computing the discrete logarithms is easy. The public file consists of a public key $y \equiv \alpha^x \bmod p$ for each user where each user has a secret x, a large prime p, and a primitive element α. To sign a document, user A uses a secret key x_A to find a signature for m in such a way that all users can verify the authenticity of the signature by using the public key y_A together with α and p, and no one can forge a signature without knowing the secret X_A.

The signature for m is the pair (r,s), $0 \leqq r,s < p-1$, chosen such that

$$\alpha^m \equiv y^r r^s \bmod p \qquad \text{Equation (1)}$$

is satisfied.

In many applications it is convenient or necessary to sign the message on-line. However, the Rivest-Shamir-Adleman system is expensive to sign on-line. The system of ElGamal, however, allows much of the computation to be done prior to going on-line since use is made of values which are not dependent upon message m. Thus, on-line signature generation is very simple in the system of ElGamal.

The signing procedure in the method taught by ElGamal includes three steps. In the first step, a random number k is chosen such that k is uniformly between 0 and $p-1$, and $gcd(k,p-1)=1$. Next, r is determined by the relationship

$$r \equiv \alpha^k \bmod p. \qquad \text{Equation (2)}$$

In view of Equation (2), the relationship which must be satisfied for determining the signature for message m, as set forth in Equation (1), may be written as

$$\alpha^m \equiv \alpha^{xr} \alpha^{ks} \bmod p. \qquad \text{Equation (3)}$$

Equation (3) may be solved for s by using

$$m \equiv xr + ks \bmod (p-1). \qquad \text{Equation (4)}$$

Equation (4) has a solution for s provided k is chosen such that $gcd(k, p-1)=1$.

In the method taught by ElGamal it is easy to verify the authenticity of the signature (r,s) by computing both sides of Equation (1) and determining that they are equal. The chosen value of k should never be used more than once. This can be guaranteed, for example, by using a Data Encryption Standard chip in the counter mode as a stream cipher to generate values of k.

It is possible to attempt two types of attacks on the signature scheme of ElGamal. The first type of attack includes attacks designed to recover the secret key x. The second type of attack includes attacks designed to forge signatures without recovering x. Some of these attempted attacks are easily shown to be equivalent to computing discrete logarithms over GF(p).

In the first type of attack attempt an intruder may try to solve t equations of the form of Equation (4) when given $\{m_i; i=1, 2, \ldots, t\}$ documents, together with the corresponding signatures $\{(r_i,s_i): i=1, 2, \ldots, t\}$. However, there are $t+1$ unknowns in this system of equations since each signature uses a different value of k. Thus, this system of equations is underdetermined and the number of solutions is large. The reason is that each value of x yields a solution for the k_i since a system of linear equations with a diagonal matrix of coefficients results. Since $p-1$ is chosen to have at least one large prime factor q, potential recovery of x mod q would require an exponential number of message-signature pairs. If any value of k is used twice in the signing, then the system of equations is uniquely determined and x may be recoverable. Thus, for the system of ElGamal to be secure, no value of k should be used more than once, as previously described.

In another attack attempt of this first type an intruder may try to solve equations of the form of Equation (3). This is always equivalent to computing discrete logarithms over GF(p), since both unknowns x and k appear in the exponent. In still another attack of this type an intruder may attempt to develop some linear dependencies among the unknowns $\{k_i 1, 2, \ldots, t\}$. This is also equivalent to computing discrete logarithms since if $k_i \equiv ck_j \bmod (p-1)$, then $r_i \equiv r^f \bmod p$, and if c can be computed then computing discrete logarithms is easy.

In the second type of attack attempt, trying to forge signatures without knowledge of x, a forger may try to find r and s such that Equation (1) is satisfied for a document m. If $r \equiv \alpha^j \bmod p$ is fixed for some j chosen at random, then computing s is equivalent to solving a discrete logarithm problem over GF(p).

If the forger fixes s first, then r may be computed as follows:

$$r^s y^r \equiv A \bmod p. \qquad \text{Equation (b)}$$

Solving Equation (5) for r may not be as hard as computing discrete logarithms. However, it is believed that solving Equation (5) in polynomial time is not feasible. In another possible attack of the second type, a forger may try to solve Equation (1) for both r and s simultaneously. However, it is believed that an efficient algorithm for doing so is not known.

The signature scheme of ElGamal also permits an attack attempt wherein the intruder, knowing one legitimate signature (r,s) for one message m, may generate other legitimate signatures (r,s) and messages m. However, this attack attempt, although implementable, does not allow the intruder to sign an arbitrary message m and therefore does not break the system. This limited ability to create acceptable message-signature pairs can be avoided by requiring m to have a certain structure. Alternatively this can be avoided by applying a one-way function H to message m before signing it. This causes a potential forger to be unable to determine a value of m which corresponds to the H(m) which was signed using the method shown below. The forger must be able to transmit such an m to the verifier, if the forgery is to be considered successful.

Given a signature (r,s) for the legitimately signed message m, then

$$\alpha^m \equiv y^r r^s \bmod p.$$

Integers A, B, and C are selected by the forger arbitrarily such that (Ar−Cs) is relatively prime to p−1. The values of r', s', m' are selected such that

$$r' = r^A \alpha^B y^C \bmod p,$$

$$s' = sr'/(Ar-Cs) \bmod (p-1),$$

$$m' = r'(Am+Bs)/(Ar-Cs) \bmod (p-1).$$

Then it is claimed that (r',s') signs the message m': The verification equation will be satisfied, since

$$\begin{aligned} y^{r'} r^{s'} &\equiv y^{r'} (r^A \alpha^B y^C)^{sr'/(Ar-Cs)} \\ &\equiv (y^{r'Ar-r'Cs+r'Cs} r^{Asr'} \alpha^{Bsr'})^{1/(Ar-Cs)} \\ &\equiv ((y^{r'} r^s)^{Ar'} \alpha^{Bsr'})^{1/(Ar-Cs)} \\ &\equiv \alpha^{(mAr'+Bsr')/(Ar-Cs)} \\ &\equiv \alpha^{m'} \end{aligned}$$

wherein all calculations are performed mod p.

As a special case, setting A=0, verifiable signatures (r',s') may be generated with corresponding messages m', without access to any signature:

$$r' = \alpha^B y^C \bmod p,$$

$$s' = -r'/C \bmod (p-1),$$

$$m' = -r'B/C \bmod (p-1).$$

Thus it will be understood by those skilled in the art that applying a one-way function H to message m, prior to signing, thwarts the general and special-case attack attempts. It will also be understood that function H may be used to form a digest of long messages so that the signature function does not have to be iteratively applied to segments of the full message m. This results in further efficiency.

U.S. Pat. No. 4,995,082, issued to Schnorr, on Feb. 19, 1991, entitled "Method for Identifying Subscribers and for Generating and Verifying Electronic Signatures in a Data Exchange System," provides a system wherein communication and verification is more efficient relative to ElGamal. Additionally, the system of Schnorr maintains the extremely efficient on-line signing capability. However, some of the desirable features of ElGamal, as well as the extensive body of experience and literature associated with the ElGamal model, are not applicable to the Schnorr model.

Thus, it is desirable to provide a system having efficiencies of on-line signing, communication, and verification which are comparable to the system of Schnorr while still maintaining compatibility with the ElGamal model and its analytical tools. In particular, it is desirable to retain the complexity of the ElGamal signature equation which enables secure use of the straightforward expression H(m), rather than simplifying the signature equation at the expense of replacing H(m) by Schnorr's $H(\alpha^k \bmod p, m)$.

SUMMARY OF THE INVENTION

A method is provided for generating and verifying a digital signature of a message m. This method requires a pair of corresponding public and secret keys (y and x) for each signer, as well as a pair of public and secret values (r and k) generated for each message by the signer. The public value r is calculated according to the rule $r=(g^k \bmod p) \bmod q$. A value s is then selected according to the rule $s=k^{-1}(H(m)+xr)$ where H is a known conventional hashing function. The message m, along with the signature (r,s) is then transmitted. When the transmitted signal is received a verification process is provided. The received values of r and s are tested to determine whether they are congruent to 0 mod q. Additionally, r is tested to determine whether it is equal to v mod q, where v is computed from r, s, m and y. For legitimately executed signatures, $v=g^k \bmod p$.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGS. **1, 2** show the digital signature algorithm of the present invention,

FIG. **3** shows a hashing algorithm suitable for use within the digital signature algorithm of FIGS. **1, 2**.

DETAILED DESCRIPTION OF THE INVENTION

Referring now to FIGS. **1, 2**, there is shown digital logorithm **10**. In digital signature algorithm **10**, the two keys in a pair of private and public keys are used respectively to generate and verify digital signatures (r,s), each of which corresponds to a transmitted message m. Using digital signature algorithm **10** the holder of a private key may generate a digital signature for message m where message m may contain any amount of data. A holder of the corresponding public key may then receive message m and verify the signature (r,s). An in-

truder who does not know the private key cannot generate the signature (r,s) of the holder of the private key for any message m and therefore signatures (r,s) cannot be forged. An intruder also cannot alter any signed message m without invalidating the signature (r,s).

If digital signature algorithm **10** is to be used effectively, a means of associating a public and private key pair with each signer is required. There must be a binding of information identifying the signer with the corresponding public key. In order to insure that each private key or secret key is held by the individual whose identity is bound to the corresponding public key, this binding must be certified by a mutually trusted third party. For example, a certifying authority may sign credentials containing the public key of a user of digital signature algorithm **10** and the identity of the user to form a certificate.

Execution of digital signature algorithm **10** of the present invention begins at start terminal **5**. A user of digital signature algorithm **10** first selects a secret value of k as shown in block **15**. The selected k is a secret integer generated by the signer for each message m. The value of k is chosen such that $0<k<q$. The k of digital signature algorithm **10** may be generated in a random or pseudo-random fashion. It will be understood by those skilled in the art that the pseudo-random generation of integer k may be performed in any conventional manner.

In block **20** of digital signature algorithm **10** a determination is made of $g^k \bmod p$. It is known in the art to determine the quantity of block **20** and transmit this quantity. However, this quantity can be quite long. Therefore, in block **25**, the quantity of block **20** is reduced to a one hundred sixty bit image by reducing it mod q as follows

$$r=(g^k \bmod p) \bmod q. \qquad \text{Equation (6)}$$

In order to generate r as set forth in Equation (6), the value g is determined as follows:

$$g=h^{(p-1)/q} \bmod p. \qquad \text{Equation (7)}$$

where h is any non-zero integer mod p such that $h^{(p-1)/q}$ is not congruent to 1 mod p. The value g may be common to all users of digital signature algorithm **10**. In Equation (6), p is a prime modulus, where $2^{511}<p<2^{512}$. The prime modulus p may be common to all users of digital signature algorithm **10**. The value q is a prime divisor of $(p-1)$, where $2^{159}<q<2^{160}$. q may also be common to all users of digital signature algorithm **10**.

Execution of digital signature algorithm **10** then proceeds to block **30** where the quantity k^{-1} mod q is determined. This value will be useful in the determination of the signature for transmission within the system of digital signature algorithm **10**. It will be understood by those skilled in the art that all of the operations performed within digital signature algorithm **10** up to and including the computation of block **30** are independent of message m. Thus, these computations may be made off-line, thereby permitting a greatly shortened on-line signing procedure.

Execution of digital signature algorithm **10** then proceeds to block **35** wherein message m is hashed. This hashing of message m performed in block **35** provides an output of one hundred sixty bits or less, denoted by H(m). Many conventional hashing algorithms, suitable for hashing message m as shown in block **35** of algorithm **10**, are known in the prior art. Additionally, it will be understood that the message to which the hashing algorithm is applied may be in an unencrypted form.

When r and k^{-1} mod q are determined as set forth in Equations (6) and (7), the value of s for message m may be determined as shown in block **40** of digital signature algorithm **10**:

$$s=k^{-1}(H(m)+xr) \bmod q. \qquad \text{Equation (8)}$$

The solution of Equation (8) of block **40** of digital signature algorithm **10** also results in a one hundred sixty bit integer. The values r and s thus determined respectively in blocks **25, 40**, constitute the signature (r,s) of message m. They are transmitted along with message m to the recipient as shown in block **45**. It will be understood that m may be transmitted in an unencrypted form. Execution of algorithm **10** then proceeds by way of off-page connector **50**.

Within digital signature algorithm **10**, each signer is provided with a secret key x, where $0<x<q$. A secret key x is fixed for all messages m transmitted by an individual user of algorithm **10**. Additionally, public key y is provided to the user holding the secret key x or secret value x, where $y=g^x \bmod p$. Prior to verifying a signature (r,s), the public key y and the identity of the signer who possesses the corresponding secret key x must be available to the recipient in an authenticated manner, where the ultimate purpose of verification is to prove that (r,s) was originally created by one who had knowledge of the value of x which corresponds to the particular value of y. If x has not been compromised, this signer is known to be the one whose identity is linked to the particular y in an authenticated manner. Additionally, the recipient must know the global values g, p and q.

Execution of algorithm **10** then proceeds by way of on-page connector **55** to start terminal **60**. After receiving message m as shown in block **65**, along with its purported signature (r,s), the recipient within the system of the present invention must verify both the received r and the received s. It will be understood therefore that within digital signature algorithm **10** the prior art kernel $g^k \bmod p$ is reduced mod q and transmitted. $g^k \bmod p$ is then recovered and verified within algorithm **10**. Thus, using the system of the present invention, the prior art $g^k \bmod p$ may be reconstructed at the receiving end rather than transmitted by the sender.

Therefore, a determination is made at decision diamond **70** of algorithm **10** whether either s or r is congruent to **0** mod q. If either r or s is congruent to **0** mod q, then execution proceeds to block **115** and the received signature (r,s) is rejected by digital signature algorithm **10**. If r and s are not congruent to 0 mod g, then the recipient proceeds with verification of the received signature (r,s) as shown in dashed verification box **75**.

Digital signature algorithm **10**, upon entering dashed verification block **75**, recovers $g^k \bmod p$ as shown in dashed recovery block **80**. It is known in the art to recover $g^k \bmod p$ after receiving a transmitted message because many prior art methods transmitted $g^k \bmod p$ without any reducing prior to transmission. Within recovery block **80**, the values of u_1 and u_2 are determined as shown in block **85**. The values of block **85** are determined as $u_1=(H(m))(s)^{-1} \bmod q$, and $u_2=(r)(s)^{-1} \bmod q$. Determination of the values u_1 and u_2 permits a determination of $g^k \bmod p$ from u_1, u_2, and y as set forth

in Equation (9). This determination is shown in block **90**. It will be understood by those skilled in the art that it is not known at this point whether the quantity recovered in block **90** is a legitimate g^k mod p. However, execution of digital signature algorithm **10** proceeds on the assumption that it is legitimate and checks this assumption.

Equation (9)

$$v = (g)^{u1}(y)^{u2} \bmod p$$
$$[= ((g^{H(m)})(y^r))^{s-1} \bmod p$$
$$= (g^{H(m)+xr})^{k(H(m)+xr)-1} \bmod p$$
$$= g^k \bmod p].$$

Within dashed checking block **95**, the recovered quantity g^k mod p of Equation (9) is checked by first determining the value of w as shown in block **100**. The value of block **100** is determined as w=v mod q. In decision diamond **105** a determination is made as to whether the received value of r is equal to the mod q reduced value of g^k mod q, where m,k,r and s satisfy the relationship set forth in Equation (8), for the given value of y. If the determination of decision **105** is affirmative, execution proceeds to verify block **110** where the signature (r,s) received in block **65** is considered verified by digital signature algorithm **10**. If the determination of decision diamond **105** is negative, execution proceeds to reject box **115** where the received signature (r,s) is rejected.

The security of digital signature algorithm **10** is dependent upon maintaining the secrecy of private keys. Users of digital signature algorithm **10** must therefore guard against the unauthorized disclosure of their private keys. In addition, the hash function p of block **35** used to determine the value of s must be selected such that it is computationally infeasible to find any message m which has a given hash value. Likewise, it should be computationally infeasible to find any pair of distinct messages m which hash to the same value.

Referring now to FIG. **3**, there is shown hashing algorithm **150**. A conventional algorithm such as algorithm **150** may be found, for example, in R. L. Rivest, "The MD4 Message Digest Algorithm," Abstracts Crypto '90, pp. 281-291. As previously described, the signature and verification processes within digital signature algorithm **10** require a secure hash algorithm which takes an arbitrary length message as input and outputs a hash value of length one hundred sixty bits or less. Hashing algorithm **150** is suitable for performing the hashing function of digital signature algorithm **10** as set forth in block **35**. It will be understood by those skilled in the art that conventional hashing functions other than hashing algorithm **150** may also be used to perform the hashing function of block **35** within digital signature algorithm **10**.

Execution of hashing algorithm **150** proceeds from block **30** of digital signature algorithm **10** and begins at start terminal **152**. Hashing algorithm **150** then receives as its input a b-bit message m to be hashed as shoWn in block **153** and operates to provide a message digest A, B, C, D as its output. The number of bits p in the message m received in block **153** is an arbitrary non-negative integer. The value of p may be zero and it need not be a multiple of eight. Furthermore, b may be arbitrarily large. The bits of message m may be described as follows:

$m_0 m_1 \ldots m_{b-1}$.

The next step of hashing algorithm **150** is padding or extending message m so that its length in bits is congruent to **448**, modulo **512**, as shown in dashed padding block **155**. Thus, message m is extended so that it is just sixty-four bits short of being a multiple of five hundred twelve bits long. Padding of message m must always be performed within hashing algorithm **150**, even if the length of message m is already congruent to **448**, modulo **512**. In the case where the length of message m is already congruent to **448**, modulo **512**, five hundred twelve bits of padding are added in dashed padding block **155**.

In the padding of message m set forth in padding block **155**, a single bit having a value of one is appended to message m as shown in block **160** within padding block **155**. Then enough zero bits are appended to message m to cause the length in bits of padded message m to become congruent to **448**, modulo **512** as shown in block **165**. The padding operation of padding block **155** is thus invertible so that different inputs yield different outputs. The padding operation of dashed padding block **155** would not be invertible if it were done only with zeros.

Execution of hashing algorithm **150** then proceeds to block **170**, where a sixty-four bit representation of p is appended to the result of the appending operations of blocks **160, 165**. It will be understood that p is the length of message m before the padding bits are added as set forth in blocks **160, 165**. This sixty-four bit representation is appended as two thirty-two bit words, low-order word first. In the unlikely event that p is greater than 2^{64}, then only the low-order sixty four bits are appended in block **170**. At this stage in the execution of hashing algorithm **150**, the resulting padded message has a length that is an exact multiple of five hundred twelve bits. Equivalently, this padded message has a length that is an exact multiple of sixteen words where each word is understood to be thirty-two bits(.) Let M[u], $0 \leqq u \leqq N-1$, denote the words of the message resulting from processing in block **170**, where p is a multiple of sixteen.

Execution of hashing algorithm **150** then proceeds to dashed message digest block **175** where a four word buffer is used to compute the message digest A, B, C, D. Each of the four words of the message digest A, B, C, D is a thirty-two bit register. In block **180** of message digest block **175** these registers are initialized to the hexadecimal values shown in Table I, low-order bytes first.

TABLE I

Word A:	01	23	45	67
Word B:	89	ab	cd	ef
Word C:	fe	dc	ba	98
Word D:	76	54	32	10

Three auxiliary functions f_1, f_2, f_3, are then defined as shown in block **185**. The auxiliary functions f_1, f_2, f_3, are set forth in Table II. Each auxiliary function f_1, f_2, f_3, of Table II receives as input three thirty-two bit words X, Y, Z and produces as output one thirty-two bit word $f_1(X,Y,Z)$, $f_2(X,Y,Z)$, and $f_3(X,Y,Z)$ respectively.

TABLE II

$$f_1(X, Y, Z) = XY \vee (\neg X)Z$$

TABLE II-continued

$f_2(X, Y, Z) = XY \vee XZ \vee YZ$
$f_3(X, Y, Z) = X \oplus Y \oplus Z$

In each bit position of the input words X, Y, Z the auxiliary function f_1 acts as a conditional to implement the condition: if X then Y else Z. In each bit position the auxiliary function f_2 acts as a majority function: if at least two of X, Y, Z have a value of one, then f_2 has a one in that bit position. The auxiliary function f_3 applies the bit-wise exclusive OR or parity function to each bit position. If the bits of X, Y, and Z are independent and unbiased, then each bit of $f_1(X,Y,Z)$ is independent and unbiased. Similarly the auxiliary functions $f_2(X,Y,Z)$ and $f_3(X,Y,Z)$ are independent and unbiased if the bits of X, Y, and Z are independent and unbiased.

Hashing algorithm **150** initializes the loop induction variable n to zero in block **186**, and then sets the current values of the array X[j] for $0 \leqq j \leqq 15$ in block **187** and performs a set of three rounds of hashing as shown in blocks **190**, **195**, **197**, where array X[j] is updated and three rounds of hashing are performed a total of N/16 times. In rounds two and three, hashing algorithm **150** uses constants. The round two constant is the square root of two and the round three constant is the square root of three. The values of these constants, with high-order digits given first, are set forth in Table III.

TABLE III

	Octal	Hex
Round 2 constant ($\sqrt{2}$)	013240474631	5A827999
Round 3 constant ($\sqrt{3}$)	015666365641	6ED9EBA1

Each of the N/16 sets of three rounds begins with execution of the instruction sequence in Table IV as occurs in block **187**, where the value of n denotes the set currently being processed. The sets are indexed by 0 to (N/16)−1.

TABLE IV

Set X[j] to M[n*16 + j], for j = 0, 1, . . . , 15.
Save A as AA, B as BB, C as CC, and D as DD.

When execution of hashing algorithm **150** proceeds to block **190** and round one of the hashing occurs, [A B C D i t] denotes the operation $A=(A+f_1(B,C,D)+X[i])<<<t$. It will be understood by those skilled in the art that (A<<<t) denotes the thirty-two bit value obtained by circularly shifting or rotating A left t bit positions. The operation denoted above generically by [A B C D i t] occurs sixteen times during round one, where the values assumed consecutively by operands A, B, C, D, I, and t respectively are given in Table V.

TABLE V

[A B C D	0	3]
[D A B C	1	7]
[C D A B	2	11]
[B C D A	3	19]
[A B C D	4	3]
[D A B C	5	7]
[C D A B	6	11]
[B C D A	7	19]
[A B C D	8	3]
[D A B C	9	7]
[C D A B	10	11]
[B C D A	11	19]
[A B C D	12	3]
[D A B C	13	7]
[C D A B	14	11]
[B C D A	15	19]

When execution proceeds to block **195**, round two of the hashing algorithm **150** begins. In round two [A B, C, D i t] denotes the operation $A=(A+f_2(B,C,D)+X[i]+5A827999)<<<t$. The operation denoted immediately above by [A B C D i t] occurs sixteen times during round two, where the values assumed consecutively by operands A, B, C, D, i, and t respectively are given in Table VI.

TABLE VI

[A B C D	0	3]
[D A B C	4	5]
[C D A B	8	9]
[B C D A	12	13]
[A B C D	1	3]
[D A B C	5	5]
[C D A B	9	9]
[B C D A	13	13]
[A B C D	2	3]
[D A B C	6	5]
[C D A B	10	9]
[B C D A	14	13]
[A B C D	3	3]
[D A B C	7	5]
[C D A B	11	9]
[B C D A	15	13]

When execution proceeds to block **197**, round three of the hashing algorithm **150** begins. In round three [A B C D i t] denotes the operation $A=(A+f_3(B,C,D)+X[i]+6ED9EBA1)<<<t$. The operation denoted immediately above by [A B C D i t] occurs sixteen times during round three, where the values assumed consecutively by operands A, B, C, D, i, and t respectively are given in Table VII.

TABLE VII

[A B C D	0	3]
[D A B C	8	9]
[C D A B	4	11]
[B C D A	12	15]
[A B C D	2	3]
[D A B C	10	9]
[C D A B	6	11]
[B C D A	14	15]
[A B C D	1	3]
[D A B C	9	9]
[C D A B	5	11]
[B C D A	13	15]
[A B C D	3	3]
[D A B C	11	9]
[C D A B	7	11]
[B C D A	15	15]

After round three is complete, execution of hashing algorithm **150** within block **35** of digital signature algorithm **10** proceeds to block **199** wherein the following additions are performed:

A = A + AA
B = B + BB
C = C + CC
D = D + DD

Thus, each of the four registers A, B, C, D which together ultimately form the digest of the received mes-

sage is incremented by the value it had before the particular set was started.

The message digest produced as the output of hashing algorithm **150** within digital signature algorithm **10** is thus the 4-tuple of values of A, B, C, D obtained in block **199** after processing the last set. The loop induction variable is incremented in block **201** and tested in decision diamond **202**. If execution is not complete block **187** is performed again. Otherwise execution of algorithm **150** proceeds to exit terminal **203**.

It will be understood by those skilled in the art that more than one hundred twenty eight bits of output may be required in some applications. This may be accomplished, for example, by providing two systems in parallel wherein each of the parallel systems executes hashing algorithm **150** with appropriately chosen constants and initialized registers, in order to provide at most two hundred fifty six bits of final output.

Although an example mode, which includes specification of parameter range restrictions, for carrying out the present invention has been herein shown and described, it will be apparent that modification and variation may be made without departing from what is regarded to be the subject matter of this invention.

I claim:

1. A method for generating a digital signature (r,s) of a message m in a system wherein information is transmitted and received by users of said system, comprising the steps of:

(a) providing a secret value k unique to said message m;

(b) providing a public value g;

(c) calculating said value r proceeding from a prime modulus p and a value g selected to be a prime divisor of p−1 according to the rule

$r=(g^k \bmod p) \bmod g$;

(d) applying a hashing transform H only to said message m to generate a transformed message H(m);

(e) calculating said value s according to the rule $s=f(H(m))$ where said value s is a function of m only by way of said transformed message H(m); and,

(f) generating a signal representative of said digital signature (r,s) in accordance with said value r and said value s and transmitting said generated signal.

2. The method for generating a digital signature (r,s) of claim **1**, wherein step (a) comprises the step of randomly selecting said secret value k.

3. The method for generating a digital signature (r,s) of claim **1**, wherein step (b) comprises the step of calculating said value g proceeding from a value h which may be any non-zero integer such that $h^{(p-1)/q}$ is not congruent to 1 mod p according to the rule

$$g=h^{(p-1)/q} \bmod p.$$

4. The method for generating a digital signature (r,s) of claim **1**, wherein step (d) comprises the step of transforming said message m by applying a one-way transform H to said message M.

5. The method for generating a digital signature (r,s) of claim **1**, wherein step (e) further comprises the step of calculating said value s according to the rule

$$s=k^{-1}(H(m)+xr) \bmod g$$

wherein said value x is a secret value.

6. The method for generating a digital signature (r,s) of claim **1**, wherein steps (a)–(c) are performed prior to knowledge of said message m.

7. The method for generating a digital signature (r,s) of claim **1**, comprising the further step of transmitting a signed message formed of said message m and said digital signature (r,s).

8. The method for generating a digital signature (r,s) of claim **7**, comprising the further steps

(g) receiving said transmitted signed message including a received digital signature (r,s) with a received value r and a received value s; and,

(h) verifying said received digital signature (r,s).

9. The method for generating a digital signature (r,s) of claim **8**, wherein step (h) comprises the step of reconstructing said $g^k \bmod p$ of step (c) to provide a recovered $g^k \bmod p$.

10. The method for generating a digital signature (r,s) of claim **9**, comprising the step of determining a value v proceeding from a value $u_1=(H(m))(s)^{-1} \bmod g$ and a value $u_2=(r)(s)^{-1} \bmod g$ according to the rule

$$v=(g)^{u1}(y)^{u2} \bmod p$$

wherein said value y is congruent to $g^x \bmod p$ and said value x is a secret value.

11. The method for generating a digital signature (r,s) of claim **10**, comprising the step of determining whether said determined value v after reduction mod q is the same as said received value r.

12. The method for generating a digital signature (r,s) of claim **11**, comprising the further step of determining that said received digital signature (r,s) is verified in response to determining that said determined value v after reduction mod q is the same as said received value r.

13. The method for generating a digital signature (r,s) of claim **8**, wherein step (h) further comprises the step of determining whether said received value r is congruent to 0 mod g.

14. The method for generating a digital signature (r,s) of claim **8**, wherein step (h) further comprises the step of determining whether said received value s is congruent to 0 mod g.

15. A system for generating a digital signature (r,s) of a message m wherein information is transmitted and received by users of said system, comprising:

a secret value k unique to said message m;

a public value g;

transform means for applying a hashing transform H only to said message m to generate a transformed message H(m);

means for calculating said value r proceeding from a prime modulus p and a value q selected to be a prime divisor of p−1 according to the rule

$r=(g^k \bmod p) \bmod q$;

means for calculating said value s according to the rule $s=f(H(m))$ where said value s is a function of said message m only by way of H(m);

generating means for receiving said calculated values of r and s and generating a signal representative of a signed message formed of said message m and said digital signature (r,s); and,

transmitting means for transmitting said generated signal.

16. The system for generating a digital signature (r,s) of claim 15, wherein said secret value k is randomly selected.

17. The system for generating a digital signature (r,s) of claim 15, wherein said public value g is calculated proceeding from a value h which may be any non-zero integer such that $h^{(p-1)/q}$ is not congruent to 1 mod p according to the rule

$$g=h^{(p-1)/q} \bmod p.$$

18. The system for generating a digital signature (r,s) of claim 15, wherein said transform means comprises one-way transform means for transforming said message m by applying a one-way hashing transform H to said message m.

19. The system for generating a digital signature (r,s) of claim 15, wherein a value x is a secret value and said value s is calculated according to the rule

$$s=k^{-1}(H(m)+xr) \bmod q.$$

20. The system for generating a digital signature (r,s) of claim 15, wherein said values k, g, and r are determined independently of said message m.

21. The system for generating a digital signature (r,s) of claim 15, further comprising:

means for receiving said transmitted signed message; and,

verifying means for verifying said digital signature (r,s).

22. The system for generating a digital signature (r,s) of claim 21, wherein said verifying means further comprises means for reconstructing said g^k mod p to provide a recovered g^k mod p within said verifying means.

23. The system for generating a digital signature (r,s) of claim 22, further comprising means for determining a value v proceeding from a value $u_1=(H(m))(s)^{-1}$ mod q and a value $u_2=(r)(s)^{-1}$ mod q according to the rule

$$\text{it } v=(g)^{u1}(y)^{u2} \bmod p$$

wherein said value y is congruent to g^x mod p and said value x is a secret value.

24. The system for generating a digital signature (r,s) of claim 23, further comprising means for determining whether said determined value of v after reduction mod q is the same as said received value r.

25. The system for generating a digital signature (r,s) of claim 24, further comprising means for determining that said signature (r,s) is verified in response to determining that said value v after reduction mod q is the same as said received value r.

26. The system for generating a digital signature (r,s) of claim 21, wherein said verifying means comprises means for determining whether said value r is congruent to 0 mod q.

27. The system for generating a digital signature (r,s) of claim 21, wherein said verifying means comprises means for determining whether said value s is congruent to 0 mod q.

28. A method for generating and verifying a digital signature (r,s) of a message m in a system, comprising the steps of:

(a) providing a secret value k unique to said message m;

(b) providing a public value g;

(c) determining said value r proceeding from a prime modulus p according to the rule $r=F(g^k \bmod p)$ wherein F is a reduction function independent of said message m;

(d) receiving a signed message formed of said message m and said digital signature (r,s);

(e) recovering and isolating g^k mod p in accordance with said message m;

(f) determining whether said isolated g^k mod p after reduction according to said reduction function F is the same as said received value r;

(g) determining that said signature (r,s) is verified in accordance with the determination of step (f); and,

(h) generating a verification signal in accordance with step (g) and transmitting said verification signal.

29. The method for generating and verifying a digital signature (r,s) of claim 28, wherein step (b) comprises calculating said value g proceeding from a value h which may be any non-zero integer such that $h^{(p-1)/q}$ is not congruent to 1 mod p according to the rule

$$g=h^{(p-1)/q} \bmod p$$

said value q being selected to be a prime divisor of p−1.

30. The method for generating and verifying a digital signature (r,s) of claim 28, wherein step (a) comprises randomly selecting said secret value k.

31. The method for generating and verifying a digital signature (r,s) of claim 29, wherein said reduction function F comprises reduction mod q.

32. The method for generating and verifying a digital signature (r,s) of claim 29, further comprising the step of determining a value v proceeding from a value $u_1=(H(m))(s)^{-1}$ mod q and a value $u_2=(r)(s)^{-1}$ mod q, according to the rule

$$v=(g)^{u1}(y)^{u2} \bmod p$$

where said value y is congruent to g^x mod p and said value x is a secret value.

33. The method for generating and verifying a digital signature (r,s) of claim 29, further comprising the step of calculating said value r proceeding from a prime modulus p, according to the rule

$$r=(g^k \bmod p) \bmod q$$

prior to knowledge of said message m.

34. The method for generating and verifying a digital signature (r,s) of claim 28, further comprising the step of calculating said value s according to the rule $s=f(H(m))$ where H is a hashing transform for producing a transformed message H(m) and said value s is a function of m only by way of said transformed message H(m).

35. The method for generating and verifying a digital signature (r,s) of claim 34, comprising the step of transforming said message m by applying a one-way transform H to said message m.

36. The method for generating and verifying a digital signature (r,s) of claim 29, further comprising the step of calculating said value s according to the rule

$$s=k^{-1}(H(m)+xr) \bmod q$$

wherein said value x is a secret value.

37. The method for generating and verifying a digital signature (r,s) of claim 36, comprising the step of determining k^{-1} prior to knowledge of message m.

38. The method for generating and verifying a digital signature (r,s) of claim 28, wherein steps (a)-(c) are formed prior to knowledge of said message m.

39. The method for generating and verifying a digital signature of claim 36, comprising the further step of transmitting a signed message formed of said message m and said digital signature (r,s) proceeding from said calculated value of s.

40. The method for generating and verifying a digital signature (r,s) of claim 29, wherein step (g) further comprises the step of determining verification in accordance with a determination whether said received value r is congruent to 0 mod q.

41. The method for generating and verifying a digital signature (r,s) of claim 29, wherein step (g) further comprises the step of determining verification in accordance with a determination whether said received value s is congruent to 0 mod q.

42. The method for generating and verifying a digital signature (r,s) of claim 5, wherein k^{-1} is determined prior to knowledge of said message m.

43. The system for generating and verifying a digital signature (r,s) of claim 19, wherein k^{-1} is determined prior to knowledge of said message m.

44. A system for generating and verifying a digital signature (r,s) of a message m wherein information is transmitted and received by user of said system, comprising:

a secret value k unique to said message m;

a public value g;

means for determining said value r proceeding from a prime modulus p according to the rule $r=F(g^k \bmod p)$ wherein F is a reduction function independent of said message m;

means for receiving a signed message formed of said message m and said digital signature (r,s);

means for recovering and isolating $g^k \bmod p$ in accordance with said message m;

comparison means for determining whether said isolated $g^k \bmod p$ after reduction according to said reduction function F is the same as said received value r;

verification means for determining that said signature (r,s) is verified in accordance with the determination of said comparison means;

means for generating a verification signal in accordance with the verification of said verification means; and,

means for transmitting said verification signal.

* * * * *

United States Patent [19]

Schnorr

[11] **Patent Number: 4,995,082**

[45] **Date of Patent: Feb. 19, 1991**

[54] **METHOD FOR IDENTIFYING SUBSCRIBERS AND FOR GENERATING AND VERIFYING ELECTRONIC SIGNATURES IN A DATA EXCHANGE SYSTEM**

[76] Inventor: **Claus P. Schnorr**, Frankfurterstr. 81, 6350 Bad Nauheim, Fed. Rep. of Germany

[21] Appl. No.: **484,127**

[22] Filed: **Feb. 23, 1990**

[30] **Foreign Application Priority Data**

Feb. 24, 1989 [EP] European Pat. Off. 89103290.6

[51] **Int. Cl.**[5] **H04K 1/00**

[52] **U.S. Cl.** **380/23**; 380/30; 380/25; 380/46

[58] **Field of Search** 380/28, 30, 25, 46, 380/23

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,225,935	9/1980	Zscheile et al.	380/28
4,351,982	9/1982	Miller et al.	380/30
4,405,828	9/1983	Rivest et al.	380/30
4,514,592	4/1985	Miyaguchi	380/28
4,658,094	4/1987	Clark	380/30
4,748,668	5/1988	Shamir et al.	380/28
4,759,063	7/1988	Chaum	380/28
4,759,064	7/1988	Chaum	380/28
4,876,716	10/1989	Okamoto	380/30

OTHER PUBLICATIONS

Omura, J. K., "A Computer Dial Access System Based on Public-Key Techniques", I.E.E.E., Communcations, vol. 25, No. 7, 1987, pp. 73-79.

Beth, T., "Efficient Zero-Knowledge Identification Scheme for Smart Cards", Advances in Cryptology--Eurocrypt, '80, pp. 77-84.

Primary Examiner—Thomas H. Tarcza
Assistant Examiner—David Cain
Attorney, Agent, or Firm—Hill, Van Santen, Steadman & Simpson

[57] **ABSTRACT**

In a data exchange system working with processor chip cards, a chip card transmits coded identification data I, v and, proceeding from a random, discrete logarithm r, an exponential value $x=2^r \pmod p$ to the subscriber who, in turn, generates and transmits a random bit sequence e to the chip card. By multiplication of a stored, private key s with the bit sequence e and by addition of the random number r, the chip card calculates a y value and transmits the y value to the subscriber who, in turn, calculates an x value from the information y, v_j and e and checks whether the calculated x value coincides with the transmitted x value. For an electronic signature, a hash value e is first calculated from an x value and from the message m to be signed and a y value is subsequently calculated from the information r, s_j and e. The numbers x and y then yield the electronic signature of the message m.

11 Claims, 3 Drawing Sheets

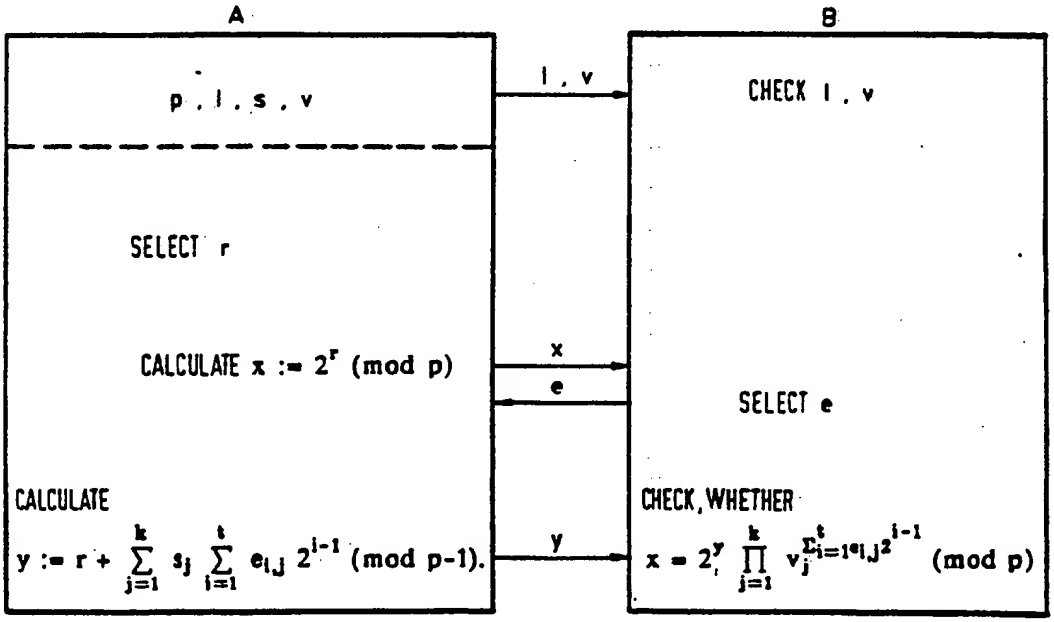


FIG 1

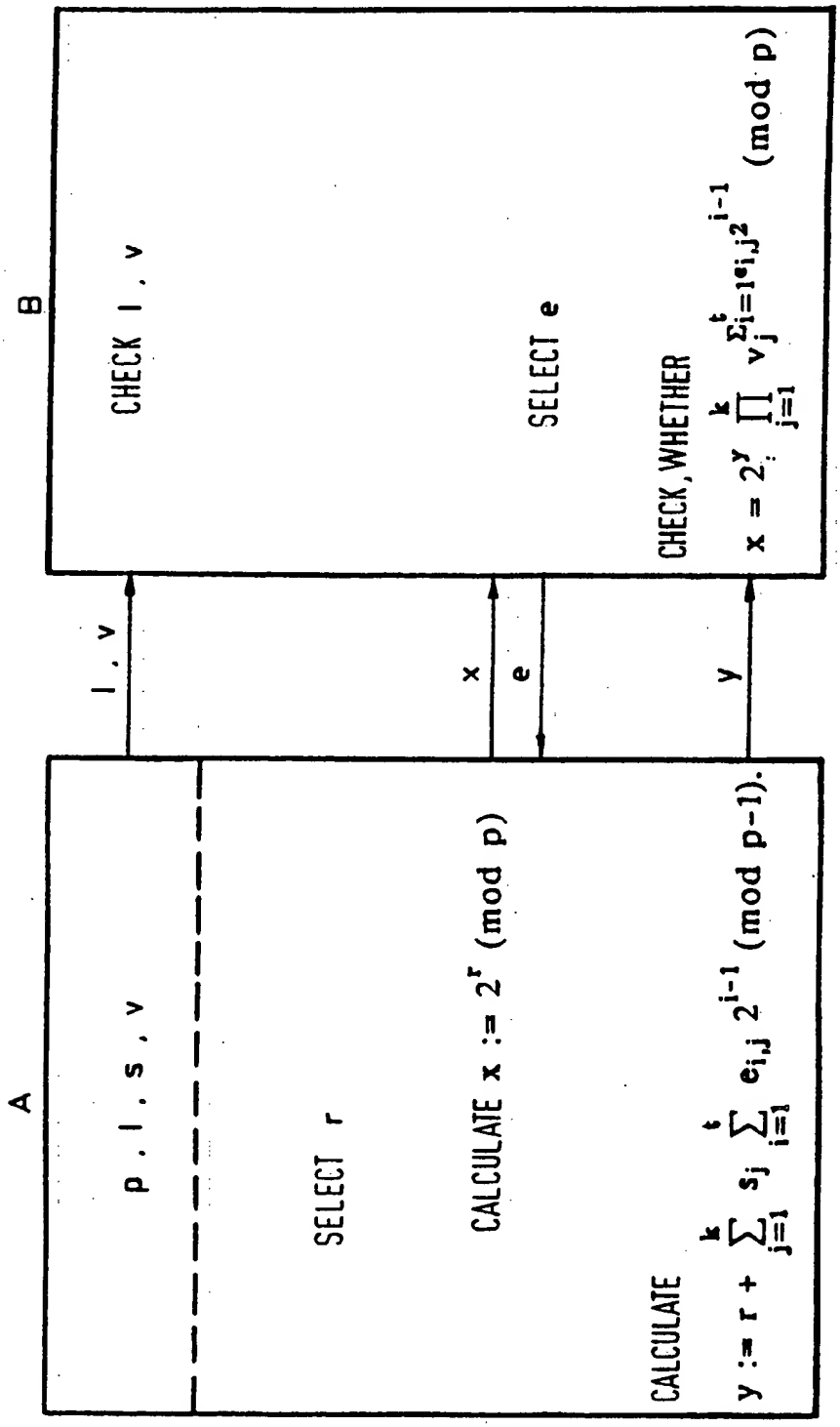

A
B
p , l , s , v
SELECT r
CALCULATE x := 2^r (mod p)
CALCULATE
y := r + Σ_{j=1}^{k} s_j Σ_{i=1}^{t} e_{i,j} 2^{i-1} (mod p-1).
l , v
x
e
y
CHECK l , v
SELECT e
CHECK, WHETHER
x = 2^y Π_{j=1}^{k} v_j^{Σ_{i=1}^{t} e_{i,j} 2^{i-1}} (mod p)

FIG 4

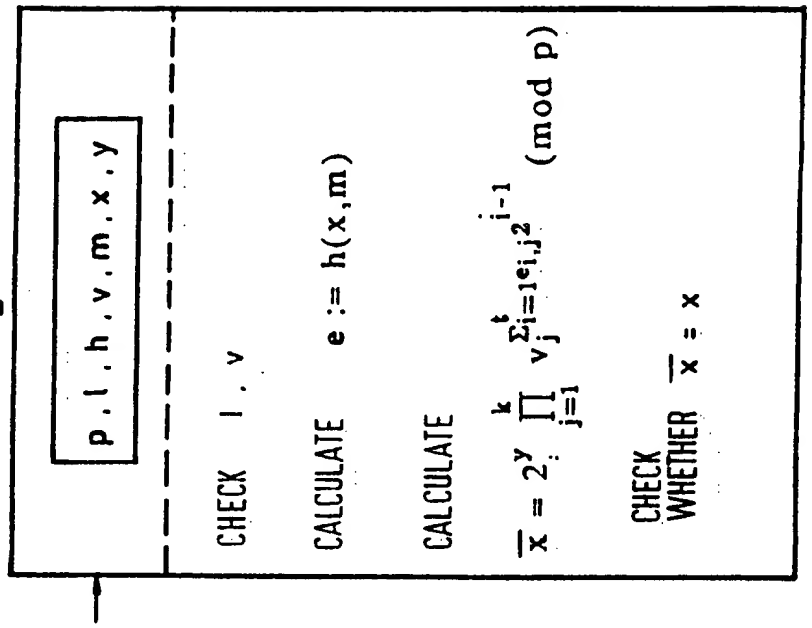

B
p , l , h , v , m , x , y
CHECK l , v
CALCULATE e := h(x,m)
CALCULATE
$\overline{x} = 2^{y} \prod_{j=1}^{k} v_j^{\sum_{i=1}^{k} e_{i,j} 2^{i-1}} \pmod p$
CHECK WHETHER $\overline{x} = x$

FIG 2

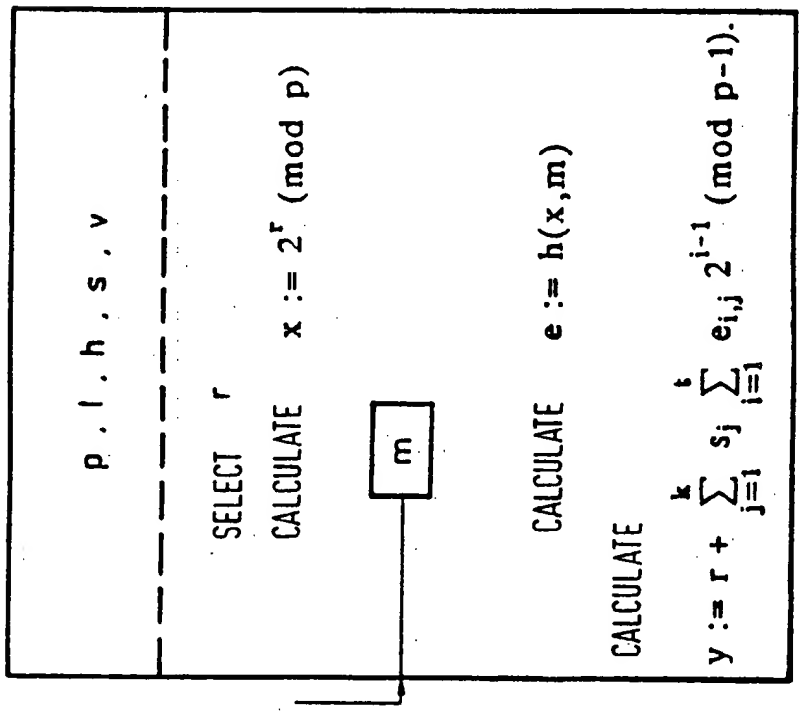

A
p , l , h , s , v
SELECT r
CALCULATE $x := 2^{r} \pmod p$
m
CALCULATE e := h(x,m)
CALCULATE
$y := r + \sum_{j=1}^{k} s_j \sum_{i=1}^{k} e_{i,j} 2^{i-1} \pmod{p-1}$.

FIG 5

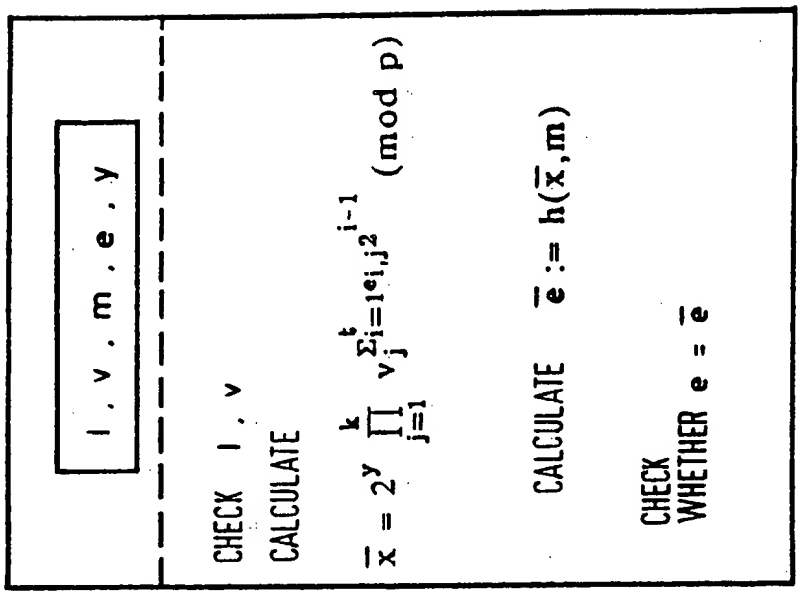
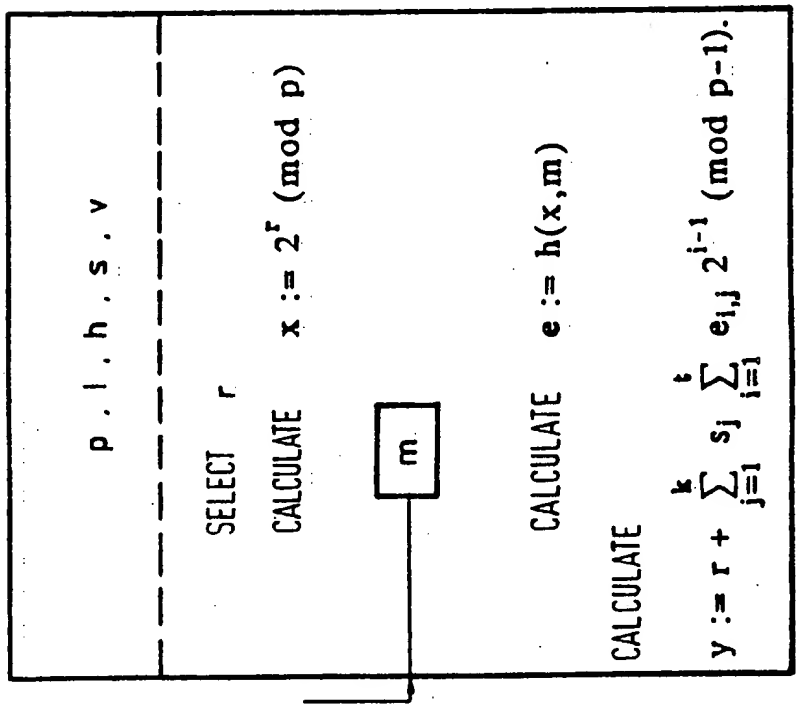

B
I , v , m , e , y
CHECK I , v
CALCULATE
$\bar{x} = 2^y \prod_{j=1}^{k} v_j^{\sum_{i=1}^{t} e_{i,j} 2^{i-1}} \pmod p$
CALCULATE $\bar{e} := h(\bar{x}, m)$
CHECK WHETHER $e = \bar{e}$

FIG 3


A
p , I , h , s , v
SELECT r
CALCULATE $x := 2^r \pmod p$
m
CALCULATE $e := h(x, m)$
CALCULATE
$y := r + \sum_{j=1}^{k} s_j \sum_{i=1}^{t} e_{i,j} 2^{i-1} \pmod{p-1}$.

METHOD FOR IDENTIFYING SUBSCRIBERS AND FOR GENERATING AND VERIFYING ELECTRONIC SIGNATURES IN A DATA EXCHANGE SYSTEM

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a method for identifying subscribers and for generating and verifying electronic signatures in a data exchange system working with processor chip cards, using identification data coded in a center with respective subscriber-related known ciphers and stored in the respective chip card and with secret ciphers having a logical relationship to the known ciphers, whereby random number-dependent check data are mutually exchanged between the subscribers.

2. Description of the Prior Art

Important prerequisites for data security in modern communication systems are:

(a) the mutual identification of the communicating partners participating in the system;

(b) the authentication of the transmitted and stored data;

(c) the coding of the transmitted and stored data; and

(d) checking the authorship of the transmitted data.

As is known, a high degree of data security can only be achieved by utilizing cryptographic methods that enable an identification and authenticity check of messages, subscribers and equipment beyond all doubt. What is generally understood by cryptography is a coding of the data for secrecy purposes. In addition to this doubtlessly-important crypto function, however, other functions, particularly checking the authenticity and authorship or generating electronic signatures are gaining increasing significance.

Symmetrical or asymmetrical coding algorithms can be employed for realizing cryptographic functions. Given a symmetrical algorithm, for example the DES algorithm (data incryption standard), identical keys are employed for coding and decoding. Symmetrical cryptosystems are particularly suitable when larger data sets have to be transmitted at a high rate. By contrast, disadvantages derive due to a relatively difficult cryptomanagement because the transmitter and the receiver must have the same key and a reliable channel is required for the transmission of the key respectively employed.

In asymmetrical cryptosystems, different ciphers are employed for coding and decoding, such that, for example, the key for coding is known and the key for decoding is secret. The latter is only known to the receiver. On asymmetrical cryptosystems, for example, the RSA algorithm named after the inventors Rivest Shamir and Adlemann that requires a comparatively high technological outlay and correspondingly long run times dependent on the length of the cipher employed but that satisfies high security requirements on the basis of the special cryptosystem. The asymmetrical cryptosystem is ideally suited for assigning a message to be transmitted. The message to be signed is thereby coded with the secret key of the signee and can be decoded by anyone that knows the public key. This "electronic signature" not only contains the personal feature (possession of private or secret key of the signee but also involves the signed text, with the consequence that the receiver recognizes any change in the text. Message and signature are therefore invariably linked via the key algorithm.

The utilization of modern cryptographic equipment is intimately connected to the introduction as what are referred to as multi-functional processor chip cards. The processor chip card not only enables versatile applications but is also employed for accepting the necessary security components (secret key and cryptoalgorithm) in order to guarantee an identification of the user and a reliable authentication of the card and of the message exchanged.

Presently known algorithms for electronic signatures, particularly the RSA algorithm (in this connection see U.S. Pat. No. 4,405,829), fully incorporated herein by this reference or the algorithm developed by A. Fiat and A. Shamir (European patent application Ser. No. 0,252,499) require either a high memory outlay or, insofar as they can be accommodated at all in the chip because of extensive and complicated arithmetic operations, particularly, multiplications, require a great deal of time, so that they are only conditionally suitable for utilization in chip cards.

SUMMARY OF THE INVENTION

It is therefore an object of the invention to provide methods for mutual identification of subscribers of data exchange systems and for generating signatures that, given essentially the same security guarantees, enable shorter run times due to more simple arithmetic operations, in comparison to known cryptographic methods.

The above object is achieved, according to the present invention, in a method for mutual identification of subscribers in a data exchange system working with processor chip cards, utilizing identification data coded in a center with respective subscriber-related known keys and stored in the respective chip card and with secret keys having a logical relationship to these known keys, whereby random number-dependent check data are mutually exchanged between the subscribers, and is particularly characterized in that the chip card sends the coded identification data, potentially together with a signature of the center, to the subscribers entering into an information exchange with the chip card, this subscriber checking the correctness of the coded identification data with reference to a known list or with reference to the signature of the center, then proceeding from a random, discrete algorithm $r \epsilon (1, \ldots, p-1)$, where p is a declared prime number modulus, the chip card forms an x value according to the rule $x := 2^r \pmod p$ and sends this x value to the subscriber, after which the subscriber sends a random bit sequence $e = (e_{l,x}, \ldots, e_{lx,k}) \epsilon \{0,1\}^{kl}$ to the chip card, and by multiplication of the stored secret key s_j that likewise represents a discrete logarithm with a binary number formed from the bits of the random bit sequence e transmitted from the subscriber to the chip card and by addition of the random number r allocated to the previously-transmitted x value, the chip card calculates a number y according to the rule

$$y := r + \prod_{j=1}^{k} s_j^{\sum_{i=1}^{l} e_{i,j} 2^{i-1}} \pmod{p-1}$$

and transmits the number y to the subscriber, then with reference to the number y transmitted to the subscriber, the subscriber calculates a number x according to the rule

$$x = 2^{y} \prod_{j=1}^{k} v_j^{\sum_{i=1}^{t} e_{ij} 2^{i-1}} \pmod{p}$$

and checks the identity of the chip card user on the basis of a comparison between the calculated number x and the x value previously communicated to the subscriber.

According to another feature of the invention, the method is particularly characterized in that the chip card calculates a x value according to the rule $x:=2^{r} \pmod{p}$ from a random number r generated in the chip card and lying in the range between 1 and the prime number modulus (p−1), that the chip card calculates a random bit sequence as a function of the x value of the message m and of a declared hash function h according to the rule $e:=h(x, m)^{\wedge}\{0,1\}^{kt}$, that the chip card calculates a y value from the random number r, from the secret ciphers s_j stored in the chip card and from the random bit sequence e according to the rule

$$y:=r+\sum_{j=1}^{k} s_j \sum_{i=1}^{t} e_{ij} 2^{i-1} \pmod{p-1}$$

and that the chip card sends the message m and the signature formed from the value x and y to the subscriber in message communication with the chip card.

According to another feature of the invention methods can be accelerated by discrete logarithms calculated in a preliminary process and intermediately stored, whereby values once employed are combined in a random fashion with other discrete logarithms in a rejuvenation process. This is exemplified by a method of the type set forth above which is particularly characterized in that a plurality of random numbers r, and respectively appertaining x values calculated in a preliminary process are stored in pairs in the chip card, in that the pair (r, x) employed in an identification procedure and/or signature procedure is varied in such a manner that a random number r, after use thereof, is combined with a random selection of the remaining stored random numbers, and in that the rejuvenated random number calculates the appertaining x value and is stored and/or used together with the rejuvenated random number r as a rejuvenated pair.

A method for verification of a signature generated according to the second-mentioned feature is particularly characterized, with respect to the subscriber receiving the signed message m, in that:

a random bit sequence e is calculated from the message m and from the x value of the signature according to the rule $e:=h(x, m)\in\{0,1\}^{kt}$,

that an x value according to the rule

$$\bar{x} = 2^{y} \prod_{j=1}^{k} v_j^{\sum_{i=1}^{t} e_{ij} 2^{i-1}} \pmod{p}$$

is calculated from the random bit sequence e, from the public key v and from the y value of the signature and is checked to see whether the calculated $\bar{x}$ value coincides with the x value of the signature.

With respect to rejuvenation, according to another feature of the invention, a method is particularly characterized in that a plurality of random numbers $r_1, \ldots, r_k$ and their appertaining x values, $x_\nu=2^{r\nu} \pmod{p}$, are stored in the chip card, and in that the pair of numbers (r, x) used in an identification procedure and/or signature procedure is rejuvenated in the following manner by a random selection $(r_{a(i)}, x_{a(i)})$ of the pairs for $i=1, \ldots, t$

$$r_\nu^{new}:=r_\nu^{old}+\sum_{i=1}^{\bar{t}} r_{a(i)} 2^{i} \pmod{p-1}$$

$$x_\nu^{new}:=x_\nu^{old} \prod_{i=1}^{\bar{t}} x_{a(i)}^{2^{i}} \pmod{p}$$

According to another feature of the invention, a method is particularly characterized by such a selection of the prime number modulus p that (p−1) is divisible by a prime number q and by such a selection of the base α of the discrete logarithm that

$$\alpha^{q}=1 \pmod{p},\ \alpha\neq 1 \pmod{p}$$

applies, and in that the discrete logarithms y, r, s_j are calculated modulo q, and in that the key components s_j and v_j are in the relationship $v_j=\alpha^{sj} \pmod{p}$. Then α plays the role of the base 2 above.

According to another feature of the invention, a method is particularly characterized by such a selection of the secret

key s_j and of the random numbers r that the bit lengths of the numbers s_j, r and y are shorter than the length of the prime number modulus p.

According to another feature of the invention, a method is particularly characterized in that other finite groups are employed for the formation of discrete logarithm instead of the finite groups that arise on the basis of residual class formation modulo p.

According to another feature of the invention, a method is particularly characterized in that a group of units Z_n° of the invertible residue classes modula a composite number n, a group of units of a finite body, an elliptical curve over a finite field or the like are provided as a finite group. Then this finite group plays the role of the group Z_p^{*}.

According to another feature of the invention, a method for verifying an abbreviated signature generated according to the third-mentioned feature at the subscriber receiving the signed message m, is particularly characterized in that:

a number $\bar{x}$ is calculated from the transmitted message m and from the signature (e, y) according to the rule

$$x = 2^{y} \prod_{j=1}^{k} v_j^{\sum_{i=1}^{t} e_{ij} 2^{i-1}}, \pmod{p}$$

and that a check is carried out to see whether the e value of the signature coincides with the value $h(\bar{x}, m)$.

The problem to be solved in practicing the present invention is comprised in the difficulty of calculating the discrete logarithm. Other, known asymmetrical cryptomethods are also constructed on this foundation (for example reference may be taken to T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE Transactions on Information Theory, Vol. 31, 1985, pp. 469–472; D. Chaum, J. H. Evertse, J. van de Graaf, "An Improved Protocol for Demonstrating Possession of Discrete Logarithms and some Generalizations", Proceedings of Eurocrypt '87, Lecture Notes in Computer Science 304, (1988), pp. 127–141; T. Beth, "A Fiat-Shamir-like Au-

thentication Protocol for the ELGAMAL Scheme", Eurocrypt '88 Abstracts, pp. 41–47). Compared to the known cryptomethods, the present invention has the advantage that the arithmetic operations can be comparatively more simply executed in the chip card. This occurs particularly due to the set preliminary process. This preliminary process can also be combined with the mentioned cryptosystems of ELGAMAL, CHAUM-EVERTSE-van de GRAAF and BETH. In addition, especially short signatures can be generated in practicing the present invention.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects, features and advantages of the invention, its organization, construction and operation will be best understood from the following detailed description, taken in conjunction with the accompanying drawings, on which:

FIG. 1 is a block diagram of the identification of a subscriber in accordance with the present invention;

FIG. 2 is an illustration of the method steps of the invention in the generating of a signature of a message to be transmitted;

FIG. 3 is a diagram of the steps for checking a signature generated according to FIG. 2;

FIG. 4 is a diagram of the method steps of the present invention in generating an abbreviated signature; and

FIG. 5 is a diagram of the steps used in the checking of the abbreviated signature generated according to FIG. 4.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

In FIG. 1, an example is illustrated how a subscriber A, for example a chip card belonging to the subscriber, proves his identity vis-a-vis a subscriber B, for example a chip card terminal.

In a data exchange system working with chip cards, the respective user-related chip cards are issued by one or, potentially, more classification centers (government representatives, credit card companies or the like), whereby the issue of the chip cards is not instituted until the identity of the respective user has been checked. The center then prepares a personal identification string I for a qualified user (name, address, ID number, etc), attaches the user-related, public key to this identification string I, this key having potentially been generated by the user himself, and publishes the pair formed of identification string I and the public key v in a publically-accessible list. The center itself does not see the secret key s and can therefore likewise not disclose the same. The identification string I, the public and secret keys v, s as well as a declared prime number p are stored in the chip card before the card is issued.

Instead of using a public list, the center can sign each pair (I,v). This signature is stored in the chip card and can be easily checked with the assistance of the public key of the center. After the chip cards and/or the public list have been issued, no further interaction with the center is necessary, neither for generating nor for checking signatures and identifications.

The identification begins with what is referred to as an initiation. The subscriber A or, respectively, the chip card thereby sends an identification string I and the public key v to the subscriber B or; respectively, to the appertaining terminal that verifies the identity. Differing from known cryptomethods, the public key is verified in the terminal, i.e. the terminal checks the relationship between the public key v and the identification string I and monitors the signature of the center in this manner. The public key $v=(v_1, \ldots v_k)$ has a logical relationship to the secret key $s=(s_1, \ldots s_k)$ and is defined as

$$v_j = 2^{-s_j} \pmod p \text{ for } j = 1, \ldots, k.$$

where p is a prime number that is at least 512 bits long. As soon as the secret key s is selected, the corresponding public key v can be easily calculated. The inverse process—calculating the secret key s from the public key v-13 cannot be implemented because the calculation of the discrete logarithm modulo p for such large prime numbers p is beyond the range of present computers and algorithms. The component s_j of the secret cipher is the discrete logarithm modulo p of f_j^{-1}, i.e.

$$s_j = -\log_2 v_j \pmod{p-1} \text{ for } j = 1, \ldots, k.$$

All discrete logarithms refer to the group ZZ^*_p (the multiplicative group modulo p) and, insofar as not otherwise noted, to the base 2. Since the order of the group Z_p^* is p−1, the discrete algorithm assumes the value 1, 2, . . .p−1. Instead of the finite groups that arise due to residual formation modulo p, other finite groups can also be employed for the formation of the discrete logarithm, such as, for example, the group of Z_n^* of invertible residue classes relative to a composite number n, the group of units of a finite field, an elliptic curve over a finite field, etc. Knowledge of the group order is not required for transferring the method to an arbitrary finite group. For example, it is adequate to calculate with the discrete logarithms on the order of magnitude of 2^{140}.

After the initiation, the subscriber A generates in record step a random number

$$r \in (1, \ldots, p-1),$$

with the corresponding exponential value $x := 2^r \pmod p$.

The inverse arithmetic process, i.e. calculating the random number r from the x value is extremely difficult insofar as p is adequately large. The subscriber B therefore has practically no possibility of discovering the random number r in the time available to him. This x value calculated at the subscriber A is transmitted to the subscriber B, i.e. to the terminal. Like the aforementioned secret key s_j, the random number r is a discrete logarithm. Following therefrom is that calculations at the side of the chip card are carried out with discrete logarithms and are carried out with the corresponding exponential value at the cooperating side, i.e. in the terminal of the subscriber B.

Generating the random number r and the exponential value

$$x := 2^r \pmod p.$$

derived therefrom can be advantageously accelerated by a preliminary process that offers and regenerates a supply of a plurality of pairs each composed of a random number r and the appertaining x value in the chip card. This supply can be set up in the chip card itself or can be externally loaded into the chip card. In an initiated identification process, one of these pairs can therefore be immediately accessed, so that the respective x

value can be immediately transmitted to the subscriber B.

In the next step, the subscriber B now sends a random bit sequence

$$e=(e_{1,1}, \ldots, e_{t,k}) \in \{0,1\}^{kt}$$

to the subscriber A or, respectively, to the chip card.

After receiving the random bit sequence e, the chip card sends a linear combination of the secret key s_j stored therein—a linear combination dependent on the bits of a random bit sequence e—, adds the current random number r thereto and transmits the numerical value y

$$y := r + \sum_{j=1}^{k} s_j \sum_{i=1}^{t} e_{ij} 2^{i-1} \pmod{p-1}$$

formed in this manner to the subscriber B.

The subscriber B now checks whether the y value sent to him is the correct answer to the question raised, the subscriber A having been asked this question by the subscriber B sending the random bit sequence e. In this check, the subscriber B calculates the right-hand part of the following equation.

$$x = 2^{y} \prod_{j=1}^{k} v_j^{\sum_{i=1}^{t} e_{ij} 2^{i-1}} \pmod{p}$$

and determines with reference to a comparison whether the calculated numerical value $\bar{x}$ coincides with the x value already previously received from the subscriber A. This task to be carried out at the subscriber B is, in fact, relatively involved; because of the adequate computer performance usually present in the terminal, it can be carried out in a relatively short time. The identification check is therefore terminated, so that the subscriber A can initiate further measures insofar as the subscriber B identified a coincidence of the two x values.

By incorporating a message m, the described identification of the subscriber A can be expanded into an electronically-generated signature of the subscriber A under the message m. This electronic signature allows the subscriber B to document the identity of the subscriber A vis-a-vis a third party, for example a judge. In addition to this, it allows the proof that the subscriber A has signed the message m beyond all doubt. The following steps must be carried out (see FIG. 2) in order to sign a message m given utilization of the secret key s_j stored at the subscriber A, i.e. in the chip card:

1. The subscriber A again selects a random number r and, as already set forth in conjunction with the identity check, calculates a x value according to the relationship

$$x := 2^{r} \pmod{p}.$$

Here also, of course, there is the possibility of accessing the stored supply and directly calling in the random numbers r and the appertaining x value.

2. The subscriber A now forms a hash value e from the message m and from the calculated x value or, respectively, from the x value taken from the supply, according to the relationship

$$e := h(x,m) \in \{0,1\}^{kt}$$

where h is thereby a publicly known hash function having values in $\{0,1\}^{kt}$.

3. Finally, the subscriber A calculates a y value from the components of the secret key s_j, random bit sequence or, respectively, hash value e and random number r according to the relationship

$$y := r + \sum_{j=1}^{k} s_j \sum_{i=1}^{t} e_{ij} 2^{i-1}. \pmod{p-1}$$

The number pair x, y then yields what is referred to as the electronic signature of the message m. The two security numbers k and t preferably lie in the range between 1 and 20. They yield a security level 2^{kt}, i.e. at least 2^{kt} multiplications (modulo p) are needed for counterfeiting the signature or, respectively, the identity. For example, k=1 and t=72 yields a security level 2^{72} that is adequate for signatures.

Proceeding on the basis of this signature formed by the number x and y, whereby both numbers are at least 512 bits long, various possibilities of abbreviating the signature derive. One of the possibilities provides that the number x be replaced by the hash value e=h(x, m) that is only 72 bits long. The signature is now composed of only y and e values (see FIG. 4). A next step is comprised in no longer taking the numbers y, r, s_j in the size of the modulo p, but of only small numbers for y, r, s_j that, however, are at least 140 bits long for the security level 2^{72}. An especially simple possibility of achieving short signatures is comprised therein that the prime number modulus p is selected such that a second prime number q divides the value (p−1), whereby q is 140 bits long. The base 2 is then replaced by a number α, so that

$$\alpha^{q} = 1 (mod\ p),\ \alpha \neq 1 (mod\ p)$$

applies. It follows therefrom that all discrete logarithms can be calculated modulo q, i.e. logarithms for the selected number α are calculated, whereby all logarithms can then lie in the range from 1through q. This has the advantage that a number that is smaller than q derives for the y value of the signature. Proceeding from the random number r

$$r \in \{1, \ldots, q-1\},$$

from

$$x := \alpha^{r} (mod\ p)$$

calculated therefrom as well as from the arbitrary bit sequence

$$e := h(x,m) \in \{0,1\}^{kt}$$

and from the number y

$$y := r + \sum_{j=1}^{k} s_j \sum_{i=1}^{t} e_{ij} 2^{i-1} \pmod{q}$$

calculated therefrom, a total length of 212 bits now derives from the signature formed from the numbers y and e with y=140 bits and e=72 bits. A signature abbreviated in this manner has the security level of 2^{72}, i.e. approximately multiplications modulo p are required in order to counterfeit a signature.

The following steps are performed by the subscriber B, i.e. in the terminal for verification of a signature composed of the numbers x and y. First, as shown in FIG. 3,

$$e:=h(x,\ m)\epsilon\{0,1\}^{kt}$$

is calculated and the equality test is then implemented such that the $\bar{x}$ value calculated according to the equation

$$x = 2^{y} \prod_{j=1}^{k} v_j^{\sum_{i=1}^{t} e_{i,j}\, 2^{i-1}} \pmod p$$

is compared to the x value of the signature.

Given abbreviated signatures in which x is replaced by e, the verification according to FIG. 5 occurs in such a fashion that

$$x = 2^{y} \prod_{j=1}^{k} v_j^{\sum_{i=1}^{t} e_{i,j}\, 2^{i-1}} \pmod p$$

is first calculated and a check is then carried out to see whether the number $\bar{x}$ supplies the correct e value. The latter occurs in that a check is carried out to see whether the hash value $h(\bar{x}, m)$ coincides with the value e.

Only relatively slight calculating tasks must be produced in the chip card both in the identification protocol and the signature protocol. Although the secret key s_j must still be multiplied by relatively small numbers in calculating the number y, this multiplication can be resolved into simple additions and shift events, what are referred to shifts, whereby the product of s_j and e_{ij} merely has to be shifted $i-1$ positions toward the left. The random number r, finally, is then to be attached to this intermediate result by addition.

Although the calculation of the number

$$x:=2^{r}(mod\ p)$$

is also involved, it can be practically neglected in terms of time expenditure due to the aforementioned preliminary process when x values corresponding to a few random numbers are calculated in advance and a plurality of pairs of numbers composed of r values and x values are stored as a supply.

In order to prevent having the same number of pairs being used over and over again at regular intervals given a limited plurality of pairs, a rejuvenation is carried out insofar as each pair, after use, is subsequently combined with other, potentially all pairs of the supply, in particular again in a random fashion. The result thereof is that the supply is rejuvenated and varied over and over, little by little.

As an example of such a rejuvenation, let it be assumed that a supply of $\bar{k}$ number pairs (r_i, x_i) is present for $i=1\ldots,k$. In order to renew the pair (r_ν, x_ν) random indices $a(1), \ldots, a(t-1)\epsilon\{1, \ldots, k\}$, for example, are selected, as is a pair (r_μ, x_μ) that has just been rejuvenated and the new pair (r_ν, x_ν) is calculated with $a(t)=\mu$ according to the rule

$$r_\nu^{new} := r_\nu^{old} + \sum_{i=1}^{\bar{t}} r_{a(i)} \pmod{p-1}$$

$$x_\nu^{new} := x_\nu^{old} \cdot \prod_{i=1}^{\bar{t}} x_{a(i)} \pmod p$$

The relationship $x=2^{r_\nu}(\text{mod } p)$ again holds true for the new pair (r_{84}, x_ν). The new number r_ν can be calculated with t additions and the new number x_ν can be calculated with $\bar{t}$ multiplication. Another rejuvenation of the pair (r_ν, x_ν) is possible according to the rule

$$r_\nu^{new} := r_\nu^{old} + \sum_{i=1}^{\bar{t}} r_{a(i)}\, 2^{i} \pmod{p-1}$$

$$x_\nu^{new} := x_\nu^{old} \cdot \prod_{i=1}^{\bar{t}} x_{a(i)} \pmod p$$

The calculation of the new value r_ν is produced here in t additions and t shifts. The new number x_ν can be calculated with 2t multiplications. Beginning with $z=1$, the steps

$$z:=zx_{a(i)}(mod\ p),\ z:=z^2(\text{mod } p),$$

are implemented for this purpose with the index i descending from t to 1. The new value x_ν is obtained as a product of the old value with the most-recently calculated number z, i.e. according to the rule

$$x_\nu^{new}:=x_\nu^{old}z(mod\ p).$$

In the rejuvenation, the selection $a(\bar{t})=\mu$ has the result that a number r_μ that was just rejuvenated is multiplied by the highest power of 2. This leads to an especially effective rejuvenation of the supply. It is advantageous to employ a pair (r, x) as a signature that is formed as a random combination of the pairs just stored. Intermediate values that arise anyway given the rejuvenation of r_ν, x_ν are well suited for this purpose.

Of course, these rejuvenation processes for the pair (r_ν, x_ν) can be combined and varied. The only matter of consequence is that the rejuvenation occurs as quickly as possible and cannot be duplicated from the signatures that have been performed. A small number $\bar{t}$ is thereby expediently employed; the rejuvenation cannot be discovered when the supply of numerical pairs—i.e. the number $\bar{k}$—is adequately large. It is advantageous to co-employ the key pairs s_j, v_j in the rejuvenation; for example, a cipher pair s_j, v_j) can be selected for a number pair $(r_{a(1)}, x_{a(i)})$. Given $\bar{t}=6$ and $\bar{k}=10$, the rejuvenation of a number pair requires only 6 or, respectively, 12 multiplications that can be implemented more or less incidently, for example when no other arithmetic operations are to be executed in the terminal.

The versatile possibilities of rejuvenating the number pairs (r_ν, x_ν) can be differently used in each chip card. For example, the indices a(1), . . . , a($\bar{t}$31 1) and the combination of the cipher pairs of the supply can be differently fashioned in each chip card. A discovery of the rejuvenation process is practically impossible in this manner.

In the case of the abbreviated signature, the random numbers r_i must be small so that the y part of the signature also remains small. This is achieved in a simple manner in that the base α for which a 140 bit long prime number q is selected for the discrete logarithms, so that $\alpha^q=1(\text{mod } p)$ is valid. The rejuvenation of the random numbers r_i, of course, is then calculated modulo q, i.e. the modulus $p-1$ is replaced by the modulus q.

Although I have described my invention by reference to particular illustrative embodiments thereof, many changes and modifications of the invention may become apparent to those skilled in the art without departing

from the spirit and scope of the invention. I therefore intend to include within the patent warranted hereon all such changes and modifications as may reasonably and properly be included within the scope of my contribution to the art.

I claim:

1. In a method for mutual identification of subscribers in a data exchange system working with processor chip cards and using identification data coded into the cards by a card-issuing center including subscriber-related public keys and stored in the respective chip cards along with private keys which have a logical relationship to the public keys, whereby random number-dependent check data are exchanged between the subscribers, comprising the steps of:

transmitting from a chip card the coded identification data together with a signature of the center to a subscriber entering into an information exchange with the chip card;

at the subscriber checking the correctness of the coded identification data with reference to known information including a public list or reference to the signature of the center;

forming in the chip card a x value proceeding from a random, discrete logarithm $r \epsilon (1, \ldots, p-1)$, where p is a declared prime number modulus, and according to the rule

$$x := 2^r \pmod p;$$

transmitting the x value to the subscriber;

transmitting from the subscriber a random bit sequence

$$e = (e_{1,1} \ldots e_{t,k}) \epsilon \{0,1\}^{kt}$$

to the chip card;

multiplying the stored, private key s_j representing a discrete logarithm with a binary number formed from the bits of the random bit sequence e transmitted from the subscriber to the chip card and adding the random number r allocated to the previously-transmitted x value to calculate, at the chip card, a number y according to the rule

$$y := r + \sum_{j=1}^{k} s_j \sum_{i=1}^{t} e_{ij} 2^{i-1} \pmod{p-1}$$

transmitting the number y to the subscriber;

at the subscriber, calculating a number x with reference to the number y according to the rule

$$\bar{x} = 2^y \prod_{j=1}^{k} v_j \sum_{i=1}^{t} e_{ij} 2^{i-1} \pmod p;$$

checking the identity of the chip card user by comparing the calculated number x and the x value previously communicated to the subscriber.

2. A method for generating a signature according to the method of claim 1, wherein:

the step of forming a x value is further defined as generating a random number r within the range of between 1 and the prime number modulus $(p-1)$ and calculating the x value according to the rule

$$x := 2^r \pmod p$$

from the generated random number r;

forming a random bit sequence as a function of the x value of a message m and of a declared hash function h according to the rule

$$e := h(x,m) \epsilon \{0,1\}^{kt};$$

calculating a y value from the random number r, from the private cipher s_j stored in the chip card and from the random bit sequence e according to the rule

$$y := r + \sum_{j=1}^{k} s_j \sum_{i=1}^{t} e_{ij} 2^{i-1} \pmod{p-1};$$

transmitting the message m and the signature formed from the value x and y to the subscriber which is in information exchange with the chip card.

3. A method for generating an abbreviated signature for a message to be transmitted in a data exchange system according to the method of claim 1, and further comprising steps defined as:

at the chip card, generating a random number r lying in the range between 1 and the prime number modulus $(p-1)$;

at the chip card, calculating a x value from the random number r according to the rule

$$x := 2^r \pmod p;$$

at the chip card, calculating a random bit sequence e as a function of the x value and of the message according to the rule

$$e := h(x,m) \epsilon (0,1)^{kt};$$

at the chip card, calculating a y value from the random number r, from the secret key s_j and from the random bit sequence e according to the rule

$$y := r + \sum_{j=1}^{k} s_j \sum_{i=1}^{t} e_{ij} 2^{i-1} \pmod{p-1};$$

transmitting from the chip card the message m and the signature formed from the values e and y to the subscriber which is information exchange with the chip card.

4. The method of claim 3, and further comprising the steps of:

generating a plurality of the random numbers r and a plurality of x values and storing the same in pairs in the chip card;

employing one of the pairs of stored random numbers r and x values (r_ν, x_ν) in an identification procedure and varying the pair in such a manner that a random number r, after use thereof, is combined with a random selection of the remaining, stored random numbers; and

calculating the appertaining x value with the rejuvenated random number and storing the same with the rejuvenated random number r as a rejuvenated pair.

5. The method of claim 4, and further defined as comprising:

storing the plurality of random numbers $r_1, \ldots r_k$ and their appertaining $x_\nu = 2^{r_\nu} \pmod p$ in the chip card; and

rejuvenating the pair (r, x) used in an identification procedure and/or a signature procedure by random selection $(r_{a(i)}, x_{a(i)})$ of the pairs for i=1, . . . , t in accordance with

$$r_v^{new} := r_v^{old} + \sum_{i=1}^{t} r_{a(i)} 2^i \pmod{p-1}$$

$$x_v^{new} := x_v^{old} \cdot \prod_{i=1}^{t} x_{a(i)}^{2^i} \pmod{p}$$

6. The method of claim 5, and further defined as:

selecting the prime number modulus p such that the number (p−1) is divisible by a prime number q and by such a selection of the base α of a discrete logarithm that

$$\alpha^q \equiv 1 \pmod{p}, \quad \alpha \not\equiv 1 \pmod{p}$$

holds true; and

calculating discrete logarithms y, r, s_j modulo q such that key components s_j and v_j are in the relationship

$$v_j \equiv \alpha^{-s_j} \pmod{p}.$$

7. The method of claim 6, and further defined as:

selecting the secret key s_j and the random numbers (r) such that the bit lengths of the numbers s_j, r and y are shorter than the length of the prime number modulus p.

8. The method of claim 6, and further defined as:

selecting finite groups for the formation of the discrete logarithm instead of the finite groups that arise on the basis of residual class modulo p.

9. The method of claim 8, and further defined as:

selecting one from the groups consisting of the Z_n^*, the group of invertible residue classes modulo q composite number r, a group of units of a finite field, and an elliptic curve over a finite field as a finite group.

10. A method for the verification of a signature (x,y) generated according to the method of claim 2 at the subscriber receiving the signed message m, comprising the steps of:

calculating a random bit sequence e from the message m and from the x value of the signature according to the rule

$$e := h(x,m) \in \{0,1\}^{kt};$$

calculating an x value according to the rule

$$\bar{x} = 2^y \prod_{j=1}^{k} v_j^{\sum_{i=1}^{t} e_{ij} 2^{i-1}} \pmod{p}$$

from the random bit sequence e, from the public cipher v and from the y value of the signature; and

comparing the calculated x value with the x value of the signature.

11. A method for verifying an abbreviated signature generated according to the method of claim 3 at the subscriber receiving the signed message m comprising the steps of:

calculating a number $\bar{x}$ from the transmitted message m and from the signature (e, y) according to the rule

$$\bar{x} = 2^y \prod_{j=1}^{k} v_j^{\sum_{i=1}^{t} e_{ij} 2^{i-1}} \pmod{p};$$

checking the value e of the signature for coincidence with the value $h(\bar{x}, m)$.

* * * * *

United States Patent [19]

Fischer

[11] **Patent Number: 4,868,877**

[45] **Date f Patent: Sep. 19, 1989**

[54] **PUBLIC KEY/SIGNATURE CRYPTOSYSTEM WITH ENHANCED DIGITAL SIGNATURE CERTIFICATION**

[76] Inventor: **Addison M. Fischer**, 60 14th Ave. South, Naples, Fla. 33942

[21] Appl. No.: **155,467**

[22] Filed: **Feb. 12, 1988**

[51] **Int. Cl.**[4] **H04L 9/00**

[52] **U.S. Cl.** **380/25**; 380/30

[58] **Field of Search** 380/23–25, 380/30

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,405,829	9/1983	Rivest et al.	380/30
4,438,824	3/1984	Mueller-Schloer	380/25
4,471,163	9/1984	Donald et al.	380/30
4,625,076	11/1986	Okamoto et al.	380/30
4,633,036	12/1986	Hellman et al.	380/30
4,759,063	7/1988	Chaum	380/30
4,759,064	7/1988	Chaum	380/30
4,771,461	9/1988	Matyas	380/25
4,799,258	11/1989	Davies	380/23
4,811,393	3/1989	Hazard	380/23

OTHER PUBLICATIONS

Recommendation X.509 pp. 63–106, "The Directory Authentication Framework", CCITT & International Standards Organization, Apr. 1988.

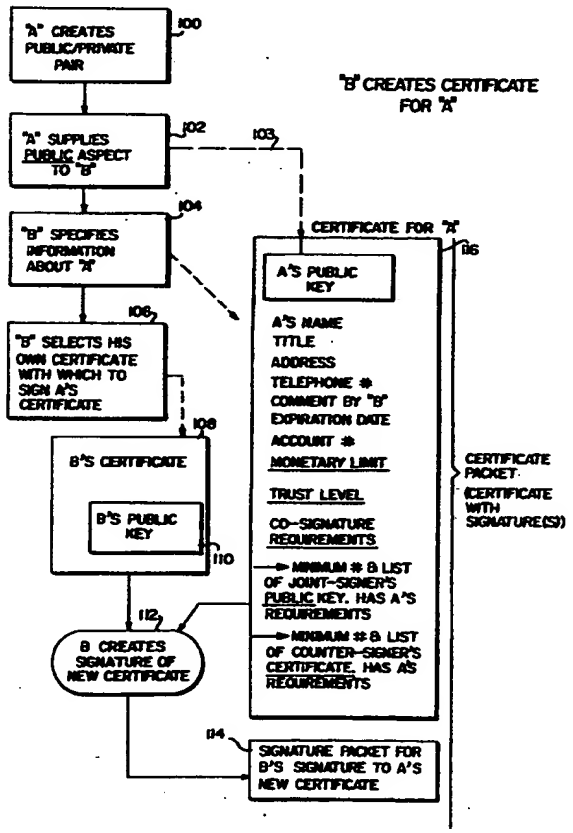
Primary Examiner—Salvatore Cangialosi

Attorney, Agent, or Firm—Nixon & Vanderhye

[57] **ABSTRACT**

A public key cryptographic system is disclosed with enhanced digital signature certification which authenticates the identity of the public key holder. A hierarchy of nested certifications and signatures are employed which indicate the authority and responsibility levels of the individual whose signature is being certified. The present invention enhances the capabilities of public key cryptography so that it may be employed in a wider variety of business transactions, even those where two parties may be virtually unknown to each other. Counter-signature and joint-signature requirements are referenced in each digital certification to permit business transactions to take place electronically, which heretofore often only would take place after at least one party physically winds his way through a corporate bureaucracy. The certifier in constructing a certificate generates a special message that includes fields indentifying the public key which is being certified, and the name of the certifiee. In addition, the certificate constructed by the certifier includes the authority which is being granted including information which reflects issues of concern to the certifier such as, for example, the monetary limit for the certifiee and the level of trust which is granted to the certifiee. The certificate may also specify cosignature requirements which are being imposed upon the certifiee.

46 Claims, 6 Drawing Sheets



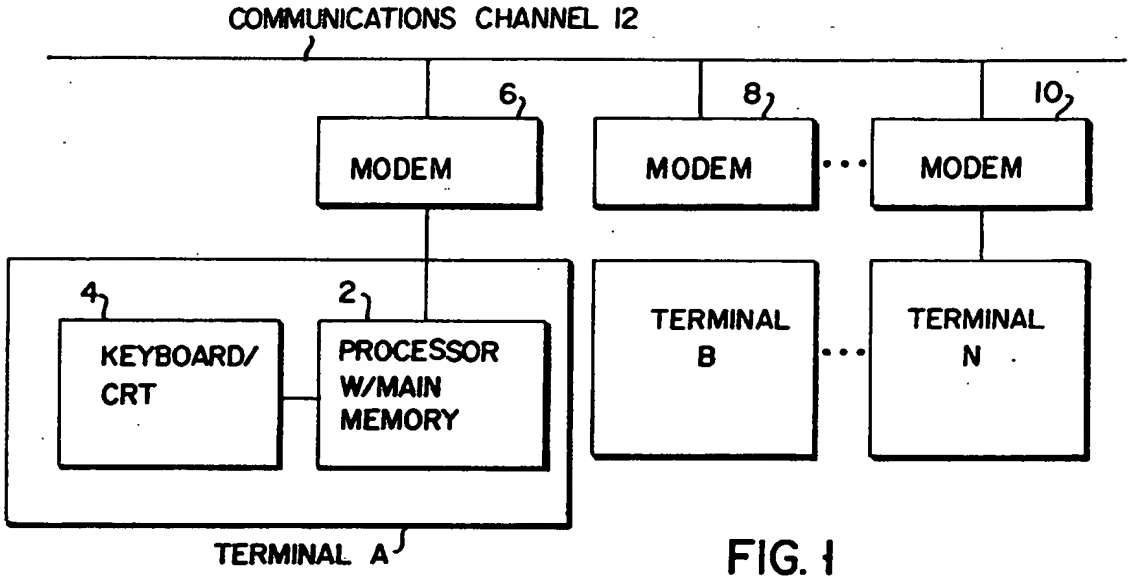


FIG. 1

FIG. 2 CREATE SIGNATURE

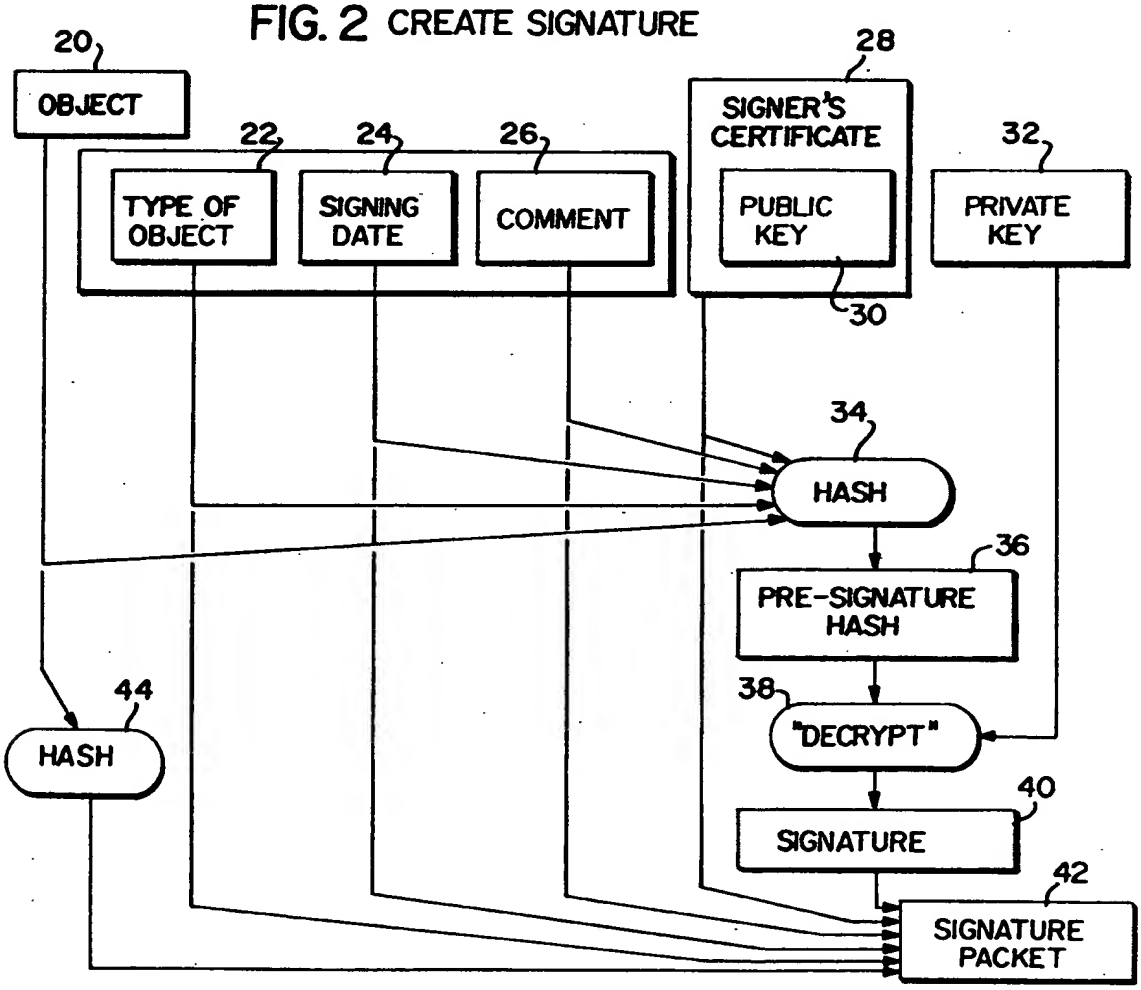


FIG. 3 VERIFY SIGNATURE

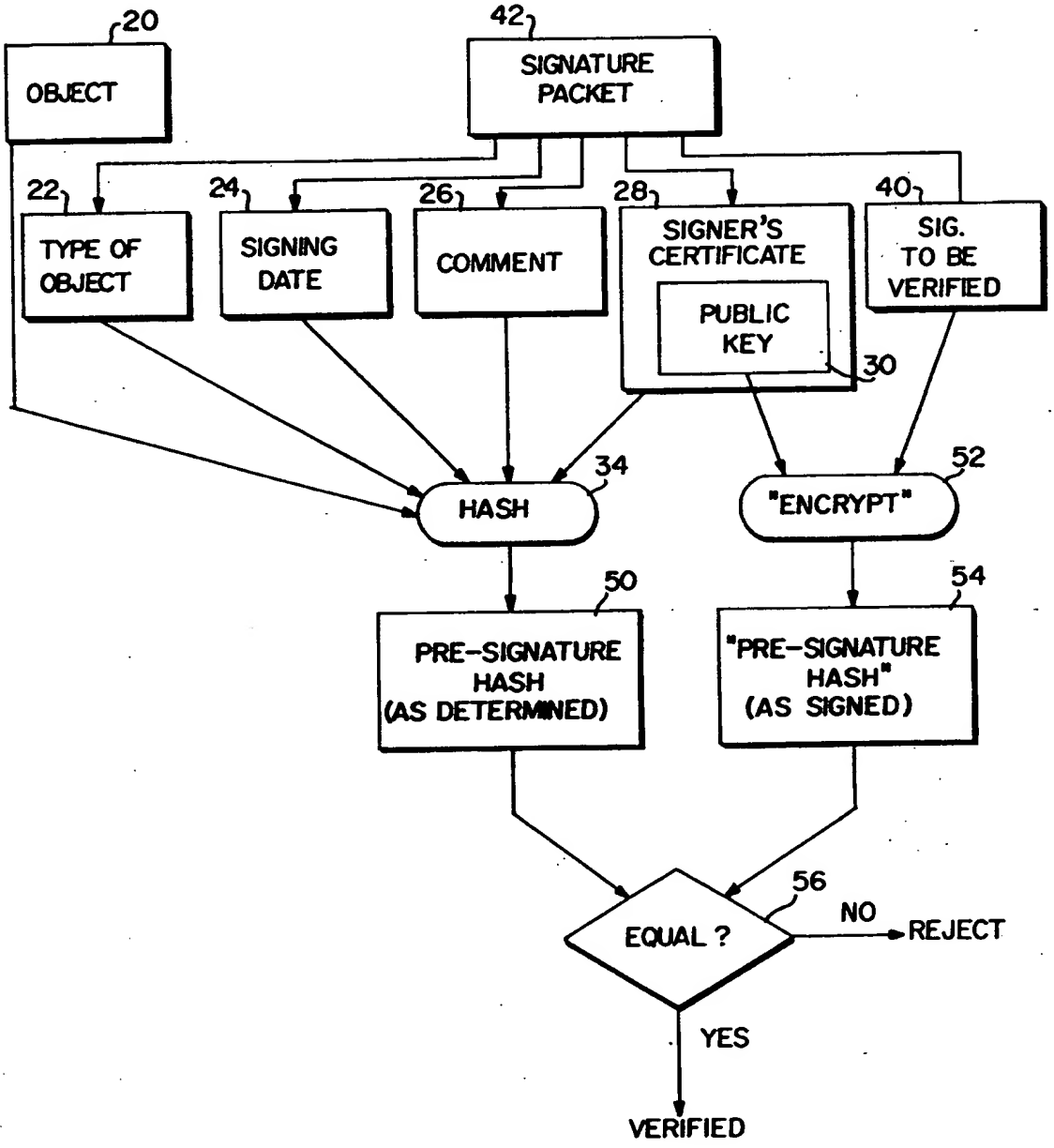
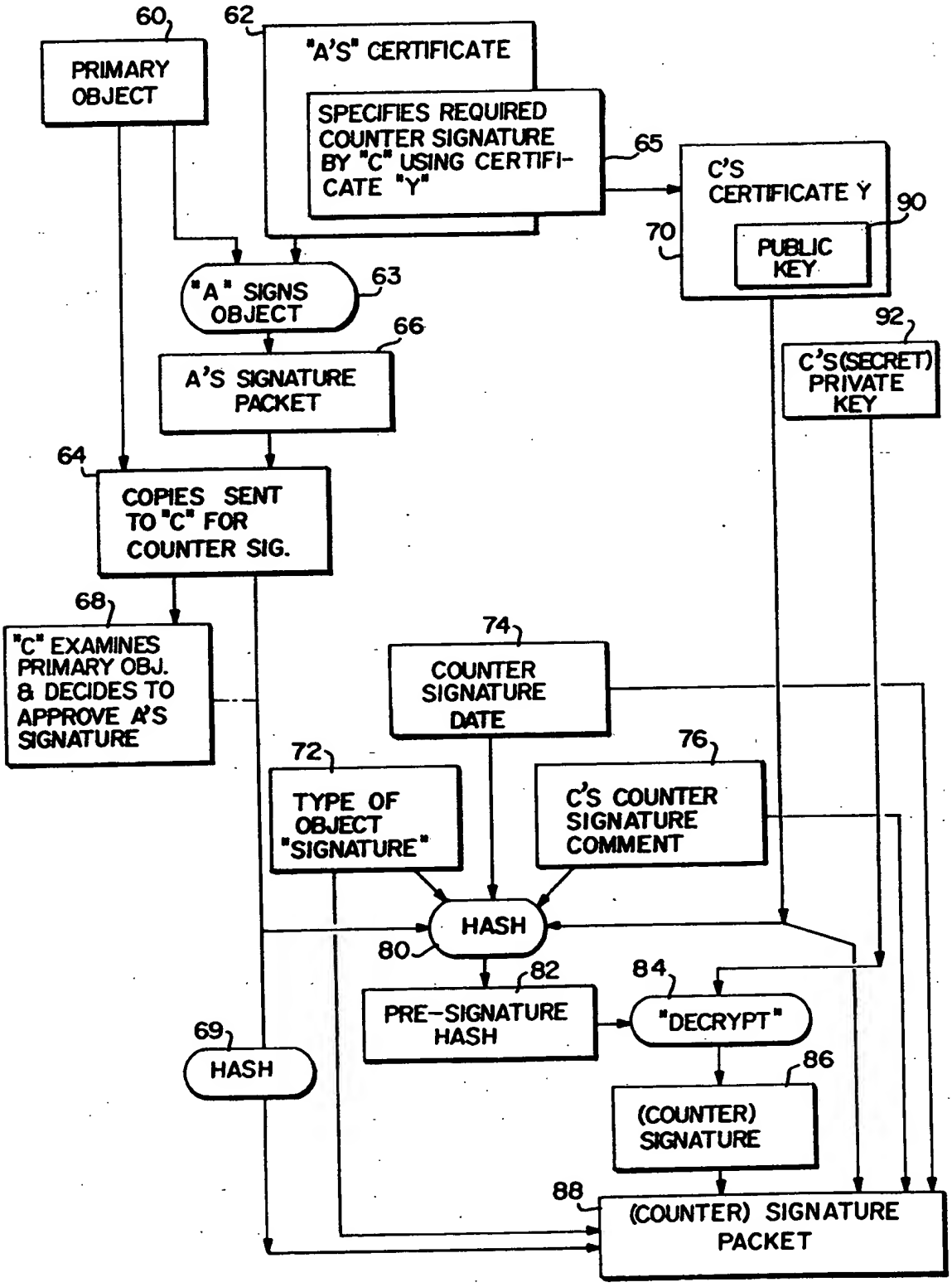


FIG. 4 "C" CREATES COUNTER-SIGNATURE FOR A'S SIGNATURE



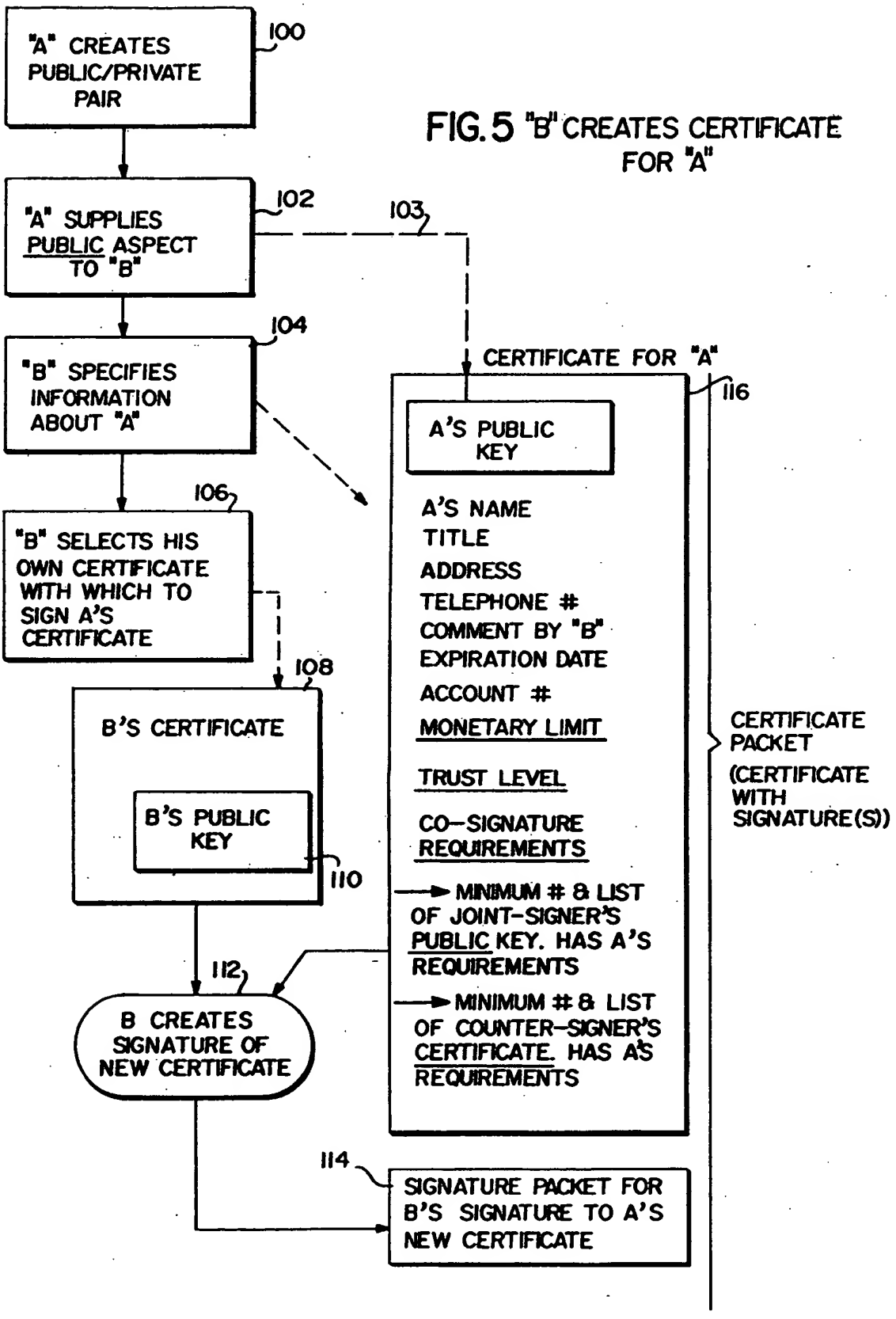


FIG. 5 "B" CREATES CERTIFICATE FOR "A"

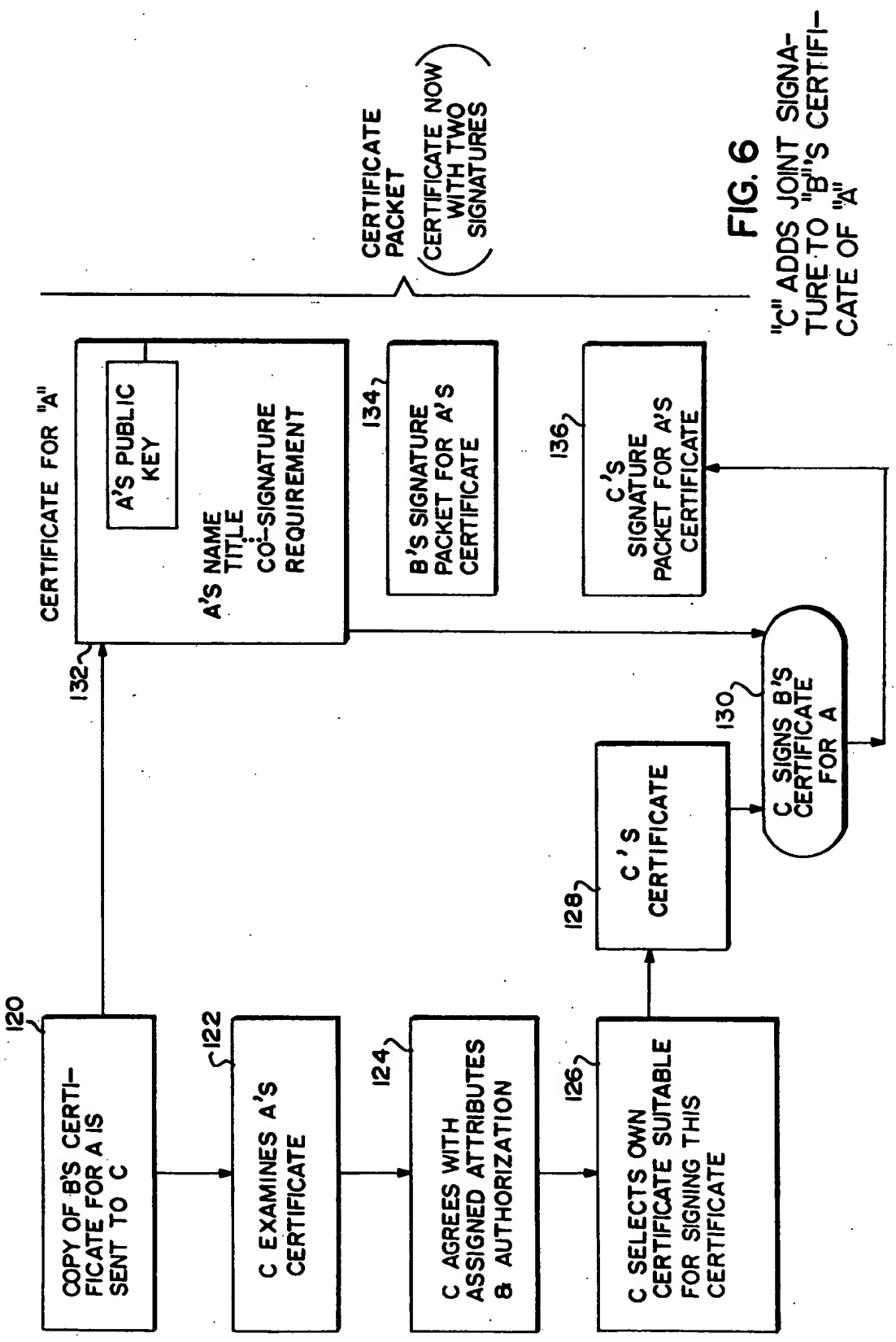

120
COPY OF B'S CERTIFICATE FOR A IS SENT TO C
122
C EXAMINES A'S CERTIFICATE
124
C AGREES WITH ASSIGNED ATTRIBUTES & AUTHORIZATION
126
C SELECTS OWN CERTIFICATE SUITABLE FOR SIGNING THIS CERTIFICATE
128
C'S CERTIFICATE
130
C SIGNS B'S CERTIFICATE FOR A
132
CERTIFICATE FOR "A"
A'S PUBLIC KEY
A'S NAME
TITLE
CO-SIGNATURE REQUIREMENT
134
B'S SIGNATURE PACKET FOR A'S CERTIFICATE
136
C'S SIGNATURE PACKET FOR A'S CERTIFICATE
CERTIFICATE PACKET
(CERTIFICATE NOW WITH TWO SIGNATURES)
"C" ADDS JOINT SIGNATURE TO "B"'S CERTIFICATE OF "A"

FIG. 6

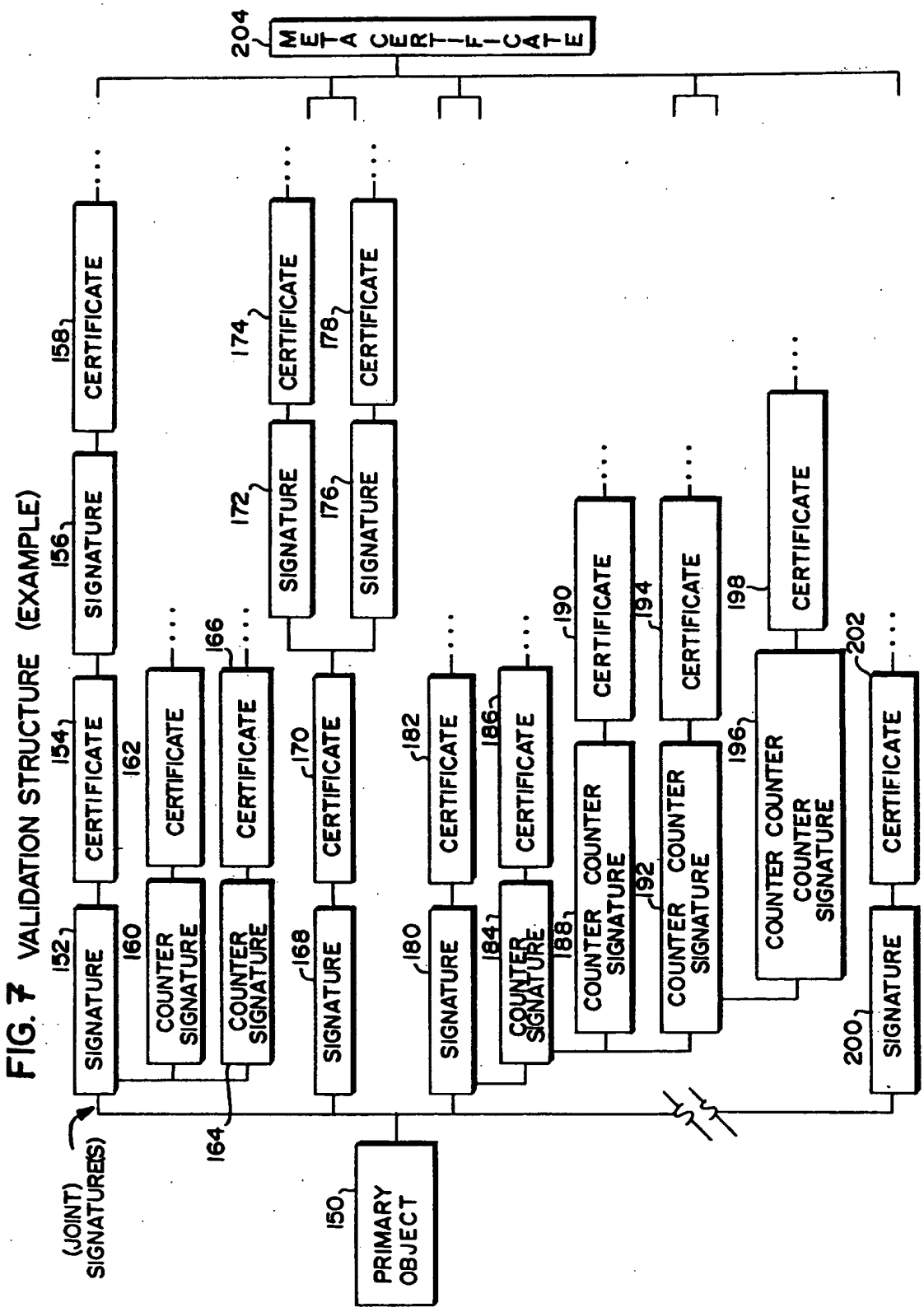


FIG. 7 VALIDATION STRUCTURE (EXAMPLE)

PUBLIC KEY/SIGNATURE CRYPTOSYSTEM WITH ENHANCED DIGITAL SIGNATURE CERTIFICATION

FIELD OF THE INVENTION

This invention relates to a cryptographic communications system and method. More particularly, the invention relates to a public key or signature cryptosystem having improved digital signature certification for indicating the identity, authority and responsibility levels associated with at least the sender of a digital message.

BACKGROUND AND SUMMARY OF THE INVENTION

The rapid growth of electronic mail systems, electronic funds transfer systems and the like has increased concerns over the security of the data transferred over unsecured communication channels. Cryptographic systems are widely used to insure the privacy and authenticity of messages communicated over such insecure channels.

In a conventional cryptographic system, a method of encryption is utilized to transform a plain text message into a message which is unintelligible. Thereafter, a method of decryption is utilized for decoding the encrypted message to restore the message to its original form.

Conventional crypotographic signatnre and authentication systems typically utilize a "one way" hashing function to transform the plain text message into a form which is unintelligible. A "hashing" function as used herein is a function which can be applied to an aggregation of data to create a smaller, more easily processed aggregation of data.

An important characteristic of the hashing function is that it be a "one-way" function. A hash is a "one-way" function, if it is far more difficult to compute the inverse of the hashing function than it is to compute the function. For all practical purposes, the value obtained from applying the hashing function to the original aggregation of data is an unforgeable unique fingerprint of the original data. If the original data is changed in any manner, the hash of such modified data will likewise be different.

In conventional cryptographic systems, binary coded information is encrypted into an unintelligible form called cipher and decrypted back into its original form utilizing an algorithm which sequences through encipher and decipher operations utilizing a binary code called a key. For example, the National Bureau of Standards in 1977 approved a block cipher algorithm referred as the *Data Encryption Standard* (DES). *Data Encryption Standard*, FIPS PUB 46, National Bureau of Standards, Jan. 5, 1977.

In DES, binary coded data is cryptographically protected using the DES algorithm in conjunction with a key. Each member of a group of authorized users of encrypted computer data must have the key that was used to encipher the data in order to use it. This key held by each member in common is used to decipher the data received in cipher form from other members of the group.

The key chosen for use in a particular application makes the results of encrypting data using the DES algorithm unique. Selection of a different key causes the cipher that is produced for a given set of inputs to be different. Unauthorized recipients of the cipher text who know the DES algorithm, but who do not have the secret key, cannot derive the original data algorithmically.

Thus, the cryptographic security of the data depends on the security provided for the key used to encipher and decipher the data. As in most conventional cryptographic systems the ultimate security of the DES system critically depends on maintaining the secrecy of the cryptographic key. Keys defined by the DES system include sixty-four binary digits of which fifty-six are used directly by the DES algorithm as the significant digits of the key and eight bits are used for error detection.

In such conventional cryptographic systems, some secure method must be utilized to distribute a secret key to the message sender and receiver. Thus, one of the major difficulties with existing cryptographic systems is the need for the sender and receiver to exchange a single key in such a manner that an unauthorized party does not have access to the key.

The exchange of such a key is frequently done by sending the key, prior to a message exchange, via, for example, a private courier or registered mail. While providing the necessary security such key distribution techniques are usually slow and expensive. If the need for the sender and receiver is only to have one private message exchange, such an exchange could be accomplished by private courier or registered mail, thereby rendering the cryptographic communication unnecessary. Moreover, if the need to communicate privately is urgent the time required to distribute the private key causes an unacceptable delay.

Public key cryptographic systems solve many of the key distribution problems associated with conventional cryptographic systems. In public key cryptographic systems the encrypting and decrypting processes are decoupled in such a manner that the encrypting process key is separate and distinct from the decrypting process key. Thus, for each encryption key there is a corresponding decryption key which is not the same as the encryption key. Even with knowledge of the encryption key, it is not feasible to compute the decryption key.

With a public key system, it is possible to communicate privately without transmitting any secret keys. The public key system does require that an encryption/decryption key pair be generated. The encryption keys for all users may be distributed or published and anyone desiring to communicate simply encrypts his or her message under the destination user's public key.

Only the destination user, who retains the secret decrypting key, is able to decipher the transmitted message. Revealing the encryption key discloses nothing useful about the decrypting key, i.e., only persons having knowledge of the decrypting can decrypt the message. The RSA cryptographic system which is disclosed in U.S. Pat. No. 4,405,829 issued to Rivest et al. discloses an exemplary methodology for a practical implementation of a public key cryptographic system.

A major problem in public key and other cryptographic systems is the need to confirm that the sender of a received message is actually the person named in the message. An authenticating technique known utilizing "digital signatures" allows a user to employ his secret key to "sign a message" which the receiving party or a

third party can validate using the originator's public key. See for example U.S. Pat. N . 4,405,829.

A user who has filed a public key in a publicly accessible file can digitally sign a message by decrypting the message or a hash of it with the user's private key before transmitting the message. Recipients of the message can verify the message or signature by encrypting it with the sender's public encryption key. Thus, the digital signature process is essentially the reverse of the typical cryptographic process in that the message is first decrypted and then encrypted. Anyone who has the user's public encryption key can read the message or signature, but only the sender having the secret decryption could have created the message or signature.

Serious problems still persist in public key cryptosystems of assuring that a specified public key is that actually created by the specified individual. One known technique for addressing this problem is to rely on some trusted authority, e.g., a governmental agency, to insure that each public key is associated with the person who claiming to be the true author.

The trusted authority creates a digital message which contains the claimant's public key and the name of the claimant (which is accurate to the authority's satisfaction) and a representative of the authority signs the digital message with the authority's own digital signature. This digital message, often known as a certificate, is sent along with the user of the claimant's own digital signature. Any recipient of the claimant's message can trust the signature, provided that the recipient recognizes the authority's public key (which enables verification of the authority's signature) and to the extent that the recipient trusts the authority.

Prior to the present invention, the transmitted certificate failed to provide any indication of the degree of trust or the level of responsibility with which the sender of the message should be empowered. Instead, the certification merely indicates that the identified trusted authority recognized the sender's public key as belonging to that person.

The public key system is designed to operate such that the public keys of various users are published to make private communications easier to accomplish. However, as the number of parties who desire to use the public key system expands, the number of published keys will soon grow to a size where the issuing authority of the public keys can not reasonably insure that the parties whose public keys are published are, in fact, the people who they are claiming to be. Thus, a party may provide a public key to be maintained in the public directory under the name of the chairman of a major corporation, e.g., for example, General Motors Corporation. Such an individual may then be in a position to receive private messages directed to the chairman of General Motors or to create signatures which ostensibly belong to the impersonated chairman.

There are also technologies for producing digital signatures which may not require full public key capability, including, for example, the Fiat-Shamir algorithm. Any digital signature methodology may be employed to implement the digital signatures referenced herein. Any reference to public key cryptosystems should also be construed to reflect signature systems. Any reference to public key decryption should be taken as a generalized reference to signature creation and any reference to encryption should be taken as a reference to signature verification.

The present invention addresses such problems with the public key or signature cryptographic system relating to authenticating the identity of the public key holder by expanding the capability of digital signature certification. In this regard, a certification methodology is utilized which employs multiple level certification while at the same time indicating the authority and responsibility levels of the individual whose signature is being certified as is explained in detail below.

The present invention enhances the capabilities of public key cryptography so that it may be employed in a wider variety of business transactions, even those where two parties may be virtually unknown to each other.

The digital signature certification method and apparatus of the present invention provides for a hierarchy of certifications and signatures. It also allows for co-signature requirements. In this regard, counter-signature and joint-signature requirements are referenced in each digital certification to permit business transactions to take place electronically, which heretofore often only would take place after at least one party physically winds his way through a corporate bureaucracy.

In the present invention, a digital signature is certified in a way which indicates the authority the has been granted to the party being certified (the certifiee). The certifier in constructing a certificate generates a special message that includes fields identifying the public key which is being certified, and the name of the certifiee. In addition, the certificate constructed by the certifier includes the authority which is being granted and limitations and safeguards which are imposed including information which reflects issues of concern to the certifier such as, for example, the monetary limit for the certifiee and the level of trust which is granted to, the certifiee. The certificate may also specify co-signature requirements as being imposed upon the certifiee

The present invention further provides for certifying digital signatures such that requirement for further joint certifying signatures is made apparent to any receiver of a digital message. The requirement for joint signatures is especially useful in transactions where money is to be transferred or authorized to be released. To accomplish this end, the certificate of the present invention is constructed to reflect (in addition to the public key and the name of the certifiee and other fields) the number of joint signatures required and an indication as to the identity of qualifying joint signers. Thus, an explicit list of each of the other public key holders that are required to sign jointly may be included in the certificate. In this fashion, the recipient is informed that any material which is signed by the authority of the sender's certificate, must also be signed by a number of other specified signators. The recipient is therefore able to verify other joint and counter signatures by simply comparing the public keys present in each signature in the certificate. The present invention also includes other ways of indicating co-signature requirements such as by indicating other certificates. Such indications of other public key holders may be explicit (with a list as described here), or implicitly, by specifying some other attribute or affiliation. This attribute or affiliation may also be indicated in each co-signer' certificate.

Additionally, the present invention provides for the certification of digital signatures such that a trust level is granted to the recipient for doing subcertifications. In this manner, a trust level of responsibility flows from a central trusted source.

In an exemplary embodiment of the present invention, a certifier is permitted to assign with one predetermined digital code a trust level which indicates that the certifier warrants that the user named in the certificate is known to the certifier and is certified to use the associated public key. However, by virtue of this digital code, the user is not authorized to make any further identifications or certifications on the certifier's behalf. Alternatively, the certifier may issue a certificate having other digital codes including a code which indicates that the user of the public key is trusted to accurately identify other persons on the certifier's behalf and is further trusted to delegate this authority as the user sees fit.

The present invention further provides for a user's public key to be certified in multiple ways (e.g., certificates by different certifiers). The present invention contemplates including the appropriate certificates as part of a user's signed message. Such certificates include a certificate for the signer's certifier and for the certifiers' certifier, etc., up to a predetermined certificate which is trusted by all parties involved. When this is done, each signed message unequivocally contains the ladder or hierarchy of certificates and the signatures indicating the sender's authority. A recipient of such a signed message can verify that authority such that business transactions can be immediately made based upon an analysis of the signed message together with the full hierarchy of certificates.

BRIEF DESCRIPTION OF THE DRAWINGS

These as well as other features of this invention will be better appreciated by reading the following description of the preferred embodiment of the present invention taken in conjunction with the accompanying drawings of which

FIG. 1 is a exemplary block diagram of a cryptographic communications system in accordance with an exemplary embodiment of the present invention;

FIG. 2 is a flow diagram that indicates how a digital signature is created in accordance with an exemplary embodiment of the present invention;

FIG. 3 is a flow diagram that indicates how a digital signature created in accordance with FIG. 2 is verified;

FIG. 4 is a flow diagram that indicates how a countersignature is created for a digital signature;

FIG. 5 is a flow diagram that indicates how a digital certificate in created in accordance with an exemplary embodiment of the present invention;

FIG. 6 is a flow diagram that indicates how a joint signature is added to a certificate; and

FIG. 7 is a flow diagram that indicates how the signatures and certificates are verified by a recipient of the transmitted message.

DETAILED DESCRIPTION OF THE PRESENTLY PREFERRED EMBODIMENT

FIG. 1 shows in block diagram form an exemplary communications system which may be used in conjunction with the present invention. This system includes an unsecured communication channel 12 over which communications between terminals A,B . . . N may take place. Communication channel 12 may, for example, be a telephone line. Terminals A,B through N may, by way of example only, be IBM PC's having a processor (with main memory) 2 which is coupled to a conventional keyboard/CRT 4. Each terminal A,B through N also includes a conventional IBM PC communications board (not shown) which when coupled to a conventional modem 6, 8, 10, respectively, permits the terminals to transmit and receive messages.

Each terminal is capable of generating a plain text or unenciphered message, transforming the message to an encoded, i.e., enciphered form, and transmitting the message to any of the other terminals connected to communications channel 12 (or to a communications network (not shown) which may be connected to communications channel 12). Additionally, each of the terminals A,B through N is capable of decrypting a received enciphered message to thereby generate a message in plain text form.

Each of the terminal users (as discussed above with respect to public key systems) has a public encrypting key and an associated private secret decrypting key. In the public key cryptosystem shown in FIG. 1, each terminal user is aware of the general method by which the other terminal users encrypt a message. Additionally, each terminal user is aware of the encryption key utilized by the terminal's encryption procedure to generate the enciphered message.

Each terminal user, however, by revealing his encryption procedure and encryption key does not reveal his private decryption key which is necessary to decrypt the ciphered message and to create signatures. In this regard, it is simply not feasible to compute the decryption key from knowledge of the encryption key. Each terminal user, with knowledge of another terminal's encryption key, can encrypt a private message for that terminal user. Only the terminal end user with his secret decrypting key can decrypt the transmitted message.

Besides the capability of transmitting a private message, each terminal user likewise has the capability of digitally signing a transmitted message. A message may be digitally signed by a terminal user decrypting a message with his private decrypting key before transmitting the message. Upon receiving the message, the recipient can read the message by using the sender's public encryption key. In this fashion, the recipient can verify that only the holder of the secret decryption key could have created the message. Thus, the recipient of the signed message has proof that the message originated from the sender. Further details of a digital signature methodology which may be used in conjunction with the present invention is disclosed in U.S. Pat. No. 4,405,829.

Before describing the details of the enhanced digital certification in accordance with the present invention, the general operation of FIG. 1 in an electronic mail, public key cryptographic context will initially be described. Initially, presume that the user of terminal A is a relatively low level supervisor of a General Motors computer automated design team who wishes to purchase a software package from a computer software distributor located in a different state. The computer software distributor has terminal N and an associated modem 10 located at his store.

The General Motors supervisor at terminal A constructs an electronic purchase order which identifies the item(s) being ordered and the address to which the items must be sent as well as other items which are necessary in a standard purchase order. It should be recognized that, although this example relates to an electronic purchase order, any aggregation of data which can be represented in a manner suitable f r processing with whatever public-key method is being used

for signatures may likewise be transmitted. In the more detailed description which follows such an aggregation of data, e.g., a computer data file, will generically be referred to as an "object".

The terminal A user, the General Motors supervisor, digitally signs the purchase order under the authority of a certificate which is appended to the transmitted message which will be discussed further below. Turning first to the supervisor's digital signature, a message can be "signed" by applying to at least a portion of the object being signed, the privately held signature key. By signing an image of the object (or a more compact version thereof known as a digest or hash of the object to be explained in more detail below) with the secret key, it is possible for anyone with access to the public key to encrypt this result and compare it with the object (or a recomputed hash or digit version thereof). Because only the owner of the public key could have used the secret key to perform this operation, the owner of the public key is thereby confirmed to have signed the message.

In accordance with the present invention, a digital signature is additionally accompanied by at least one valid certificate which specifies the identity of the signer and the authorization which the signer has been granted. The certificate may be viewed as a special object or message which specifies the identity of the user of a particular public key and the authority which has been granted to that user by a party having a higher level of authority than the user.

To be valid a certificate must be signed by the private key(s) associated with one or more other valid certificates which are hereafter referred to as antecedents to that certificate. Each of these antecedent certificates must grant the signer the authority to create such a signature and/or to issue the purchase order in our example. Each of the antecedent certificates may in turn have its own antecedent(s).

An exemplary embodiment of the present invention contemplates utilizing an ultimate antecedent certificate of all certificates, which is a universally known and trusted authority, e.g., hypothetically the National Bureau of Standards, and which is referred to as a meta-certificate. The meta certificate is the only item that needs to be universally trusted and known. There may be several meta-certifiers, and it is possible that meta-certificates may even reference each other for required co-signatures.

Turning back to our example, when the message is ultimately transmitted from terminal A to the computer software distributor at terminal N, the recipient in a manner which will be described in detail below, verifies the signature of the General Motors supervisor. Additionally, he verifies that all the other signatures on the message certificate and the antecedent certificates are present which provides further assurance to the terminal N software distributor that the transaction is a valid and completely authorized. As should be recognized, such assurances are critically important prior to shipping purchased items and are perhaps even more important in an electronic funds transfer context.

Any party who receives a message transmitted by the user of terminal A (whether such a party is the ultimate recipient of the message at terminal N or other parties within for example a corporate hierarchy such as General Motors) can verify and validate A's signature and the authority that the terminal A user exercised. Such validation is possible since a complete hierarchy of validating certificates is transmitted with the original purchase order which permits the ultimate recipient to feel confident that the requested transaction is authentic and properly authorized.

Focussing more generically on major transactions emanating from, for example, General Motors Corporation, it is helpful to focus first on the ultimate certifier(s) mentioned above, i.e., the meta-certifiers. In this regard, General Motors and parties who plan to do business with General Motors or otherwise participate in the public key cryptosystem may initially choose to approach a universally recognized trusted authority e.g., hypothetically the Bureau of Standards and/or one of the country's largest banks. Corporate and other participants in this system register a set of public keys (which they are authorized to use by virtue of an action of their corporate board of directors) with the meta-certifier. These are "high level" keys to be used within the General Motors environment primarily for certifying General Motors' internal personnel. The meta-certifier in return distributes to General Motors its certification that each of these supplied public keys created by General Motors is authorized for their own use. In effect, the meta-certifier is certifying that the party using each key is actually associated with General Motors. The meta-certifier's certification may include embedded text which indicates that the users of registered public keys are properly associated with General Motors. For example, General Motors may decide to have three "high level" keys certified, e.g., corporate officers, such as the vice president, financial officer, and the security officer. At General Motors' request each of the three certificates indicate the public keys of the other two as required joint signatures.

Thus, once having obtained the highest level certificate(s) from the meta-certifier, several officials within General Motors may have to jointly sign certificates at the next lower level and such joint signatures. Each of these high level General Motors' certificates would mutually reference each other as required co-signers At this level no single corporate officer acting alone may authorize anything because embedded within each of the three certificates is a requirement for the signature of others who are specifically identified. In turn then, these 3 officers create and sign public keys for the other General Motors' employees, that define exactly the level of authority, responsibility and limitations each employee is to have. One of these certificates may belong to user A, or will be an antecedent to user's A's certificate.

Each of these three high level certificates may digitally sign terminal B user's certificate preferably after a face to face or telephone verification. After each of the required signatures has been created, the certificate's signatures by the vice president, financial officer and security officer as well as their respective 3 certificates, as well as those certificates' respective signatures by the meta-certifier are ultimately returned to the General Motors' supervisor at terminal B to be stored for ongoing use, such as in our example for subcertifying terminal user A. In this manner, the signed message unequivocally contains the ladder or hierarchy of certificates and signatures verifying terminal A user's identify and his authority.

When a party B in a ladder of certifications creates an authorizing certificate for party A, the certificate includes a specification of A's identity together with A's public encryption key. Additionally, the certificate indicates the authority, capabilities and limitations which B

wishes to grant A. By granting this certificate B explicitly assumes responsibility for both A's identity and authority.

B's certificate for A also permits a specification of other parties who are required to cosign actions taken by A when using this certificate as will be explained further below. Cosignatures may take the form of either joint signatures or countersignatures. Additionally party B can define in the certificate for A the degree to which B will recognize subcertifications performed by A.

In accordance with an exemplary embodiment of the present invention, trust levels which are granted by the certifier to the certifiee are specified in the certificate by a predetermined digital code. Such a trust level is used by the recipient of the message as an indicator of the authority granted to the certifiee and the responsibility assumed by the certifier for the certifiee's actions with respect to the use of the public key being certified.

By way of example only trust levels may be indicated by trust level values 0, 1, 2, and 3.

Trust level 0 indicates that the certifier vouches that the certified public key belongs to the individual named in the certificate; but that the certifier will not assume responsibility for the accuracy of any certificates produced by the certifiee. The essence of this would be a statement by the certifier that: "I warrant the user named in this certificate is known to me and is being certified to use the associated public key—however I do not trust him to make any further identifications on my behalf".

Trust level 1 empowers the certifiee to make level 0 certifications on behalf of the certifier. The essence of this would be a statement by the certifier that: "I know the user of this public key and I trust him/her to accurately identify other persons on my behalf. I will take responsibility for such identifications. However, I do not trust this person to identify persons as trustworthy."

Trust level 2 empowers the certifiee to make level 0, 1 and 2 certifications on behalf of the certifier. The essence of this would be a statement by the certifier that: "I know the user of this public key and I trust him/her to accurately identify other persons on my behalf, and I furthermore trust them to delegate this authority as they see fit. I assume due responsibility for any certifications done by them or any duly authorized agent created by them or by other generation of duly created agents".

Trust level 3 is reserved exclusively for an ultimate meta certifier whose public key and certificate is established and also well known (possibly by repetitive and widespread media publication) and whose accuracy is universally respected. This certifier takes responsibility only for accurately identifying the entities whose public keys it certifies. It assumes no responsibility for the use of these keys.

Additionally, each certification may specify the monetary limit, i.e., the maximum amount of money value which the certifiee is authorized to deal with. The monetary limit must not of course exceed the limit in the certifier's own certificate to insure that the certifier does not delegate more than he is allowed to handle.

Before discussing further details of the digital signature and certification techniques of the present invention, it may be helpful to first define certain terminology. As noted above, the term "object" is generically used to describe any aggregation of data that can be ultimately represented in a manner suitable for processing with whatever public key method is being utilized for signatures and/or encryption. The term object may apply to a "primary" object such as a purchase order or check, or money transfer; or to a "secondary" object such as a certificate, or another signature.

The methodology of the present invention in order to increase processing efficiency generally applies a function to the object to create a generally smaller, more compact, more easily processed object, i.e., typically a fixed size bit string of several dozen or more bits. Such a function is referred to as a hash or digest of the object.

An example of such a hash or digest would be the output obtained by processing an image of the object with the data encryption standard (DES) using cipher block chaining mode (CBC). Processing may be done with two different DES keys (both of which are fixed, non-secret and commonly known). Thereafter, each of the final output chaining values are concatenated or merged in some way to become the several dozen or more bits constituting the digest or hash value.

An important characteristic of the digest or hashing algorithm is that, while it is easy to compute the digest of an object it is essentially impossible to construct a different or modified object with an equal digest. For all practical purposes the digest is an unforgeable unique fingerprint of the original object. If the original object is changed in any manner, the digest will be different. In other words, for all practical purposes, the more compact representation of the original object is unique to the original object. Ideally, also a hash should not reveal any clue about specific data values contained within the message. The hash's contemplated in the exemplary embodiment have at least 128 bits.

Turning now to FIG. 2, this figure shows the data flow and the manner in which signatures are created. The signature process applies not only to general objects such as arbitrary computer files, letters, electronic purchase orders, etc., but also to specialized objects such as signatures and certificates.

Each digital signature is accompanied, as is generally shown in FIG. 2, by a certification of the public key performing the signature. The certificate, as will be discussed in detail in conjunction with FIG. 5, is signed by one or more higher authorities (i.e., the immediate certifiers) and identifies the original signer while specifying the degree of authority which has been granted to the original signer.

In accordance with the present invention, the original signer may have more than one certificate and may utilize different certificates for different levels of authority. Each of the certificates may carry different limitations and requirements including different money limitations, trust levels, joint signature requirements and counter signature requirements.

It is incumbent on the signer to select the appropriate signature/certificate with which to sign a particular object. For example, purchase orders may require a different type of authority (and therefore a different certificate) than merely a letter of inquiry. Thus, the certificate is a very important portion of the transmitted message in that it identifies the signer as well as the signer's level of authority.

As shown in FIG. 2, in creating the signature the user utilizes the object 20 (which may, for example, be a purchase order) and specifies the type of object 22. The documentation added under the type of object field, for example, indicates that the object is a purchase order

data file. In other instances the type of object field 22 would identify that the object is another signature or a certificate. As indicated at 24, the date of the signature is also identified.

The comment field 26 is utilized to add documentation which, for example, places limitations on the signature or adds other commentary. The signer may indicate that his signature or the object is only good and valid for a predetermined period of time. Additionally, any desired comments regarding the particular transaction, e.g., the purchase order, may be added as comment data.

Also incorporated in the signature is the original signer's certificate 28 which includes the original signer's public key 30 and numerous other fields which are discussed in detail below in conjunction with FIG. 5. As noted above, public key signature methods require the use of a public key 30 and an associated private key which is shown in FIG. 2 at 32.

The object field 20 (e.g., purchase order data), the type of object field 22, the signing date field 24, the comment field 26, and the signer's certificate field 28 are hashed via a hashing algorithm at 34 to enhance processing efficiency. Additionally, each of the fields 20, 22, 24, 26 and 28 are incorporated in the signature packet 42 to become part of the signature record. A hashing algorithm 44 is also applied to the object 20 to place it in a more compact form prior to incorporation in the packet 42.

After application of the hashing algorithm 34 to the fields previously discussed, a presignature hash results therefrom as indicated at 36. The presignature hash 36 is then run through a decrypt (signature) cycle as indicated at 38 using the signer's private key 32 to thereby result in the signer's signature 40. The signer's signature 40 together with items 20 (or the hash of 20), 22, 24, 26 and 28 become the final signature packet 42.

When this signature is transmitted with the associated object, it allows the recipient to verify that the object is intact as it was signed. Furthermore, when sufficient certificates are also included, the recipient can validate the true identity of the signer and the authority which has been granted in the signer's chain of certificates.

Turning now to FIG. 3, this figure shows how a recipient of the transmitted message, including the signature packet 42 constructed in accordance with FIG. 2, verifies the signature. As shown in FIG. 3, the recipient utilizes the signature packet 42 and the associated fields 22, 24, 26 and 28 as well as the object 20 and applies the same hashing algorithm 34 as applied to these same fields in FIG. 2 to thereby result in a presignature hash 50.

The recipient then utilizes the public encrypting key transmitted with the signer's certificate 28 and performs an encrypt (verification) operation 52 on the signature to be verified 40 (which was transmitted with the signature packet) to thereby generate a presignature hash 54. The recipient, by recomputing the presignature hash in the same way as the signer, then compares this value with the encryption (verification) of the signer's signature.

As indicated at block 56, if these two values at 50 and 54 are not equal, the recipient cannot accept the received signature as being valid. Whether intentional or otherwise, the object and/or the signature must have been changed or tampered with in some way since they were signed. By virtue of this verification step, the recipient determines that the digital signal is consistent with the public key that was named.

In this manner, the object and its signature are processed to insure that the object is identical to the data which existed as it was signed by the owner of the public key. This is the first step of an overall validation process.

Other steps in the validation process insure that the public key belongs to the person named in the associated certificate and that the person has the authority stipulated in the certificate. This verification process applies generally to any object even if that object is another signature or a certificate. To complete the validation process, the recipient analyzes the certificates associated with the signature to determine that the proper authority has been conveyed to each certificate through its signatures and the antecedent certificate(s) of these authorizing signatures.

An object may be accompanied by more than one signature. Such cosignatures fall into the category of either a joint signature or a counter signature. A joint signature is simply another signature of an object by a different party. The signature process is no different than that used to create the initial signature as described in conjunction with FIG. 2.

A counter signature is a signature of a signature. Thus, when A signs an object, this signature may itself be thought of as an object. Thus, when C countersigns A's signature, the object C is signing is A's signature itself rather than the original object. The counter signature must therefore occur after the signature being countersigned and reflects approval (or at least recognition) of both the underlying object as well as the fact that A has signed that object. This mechanism allows a chain of authority where each higher level approves any commitment made at a lower level. One of the unique aspects of this system is that the certificate A associates with this signature may in fact require that the use of A's signature be accompanied by particular other joint or counter signatures.

Turning next to the creation of a counter signature which is shown in FIG. 4, initially A signs at 63 a primary object 60 in accordance with the procedure outlined in detail in conjunction with FIG. 2. The primary object 60 may be a purchase order or some other commitment or it may be a counter signature of some other signature of a primary object.

As explained above in regard to FIG. 2, the process of A signing an object may also involve some other party signing A's signature. Thus, A's certificate 62 specifically defines at 65 that, in order for A's signature to be valid (i.e., ratified), a counter signature by C is required, for example, using C's specific certificate Y.

After A signs the object, A's signature packet 66 is then forwarded along with the primary object and all associated signatures and certificates to C and A requests that C add his counter signature 64. Upon receiving the material, C reviews all existing signature certificates and the primary object and if everything meets with his approval he would decide to sign A's signature 68. A's signature inherently reflects the primary object and C's signature inherently reflects A's signature so C will essentially have "signed on the line below A's signature".

Once C decides to approve A's signature at 68, the process of creating a signature as shown in detail in FIG. 2, is duplicated except that the object is A's signature. Thus, with A's signature as the object (and the

type of object being designated as a signature at 72), the counter signature date 74, C's counter signature comment 76, and C's certificate 70 are applied to a hashing algorithm 80 to thereby result in a presignature hash 82. At the same time, these fields are also inserted into the counter signature packet 88 as discussed above with respect to the signature packet 42 (with a hashing algorithm 69 being applied to the signature object).

Presignature hash 82 and C's secret key 92 are applied in a signature operation 84 to generate a counter signature 86. This counter signature becomes part of the counter signature packet 88 in precisely the same fashion as described previously in regard to the creation of signature packet 42 in FIG. 2.

Because the certificate "Y" which C must use to perform the signature has been explicitly stated (in the certificate which A used to sign), C may also be required to meet his own cosignature obligations so specified by "Y" and forward this entire package including his own newly added signature on to other parties for further cosignatures (either joint or counter signatures). This recursive signature gathering process continues until sufficient signatures are added to fully satisfy all cosignature requirements of at least one party who initially signed the primary object.

Turning next to how one party creates an authorizing certificate for another, it is noted that B creates an authorizing certificate for A by combining a specification of A's identity together with the public encrypting key which A generated for himself. Additionally B specifies the authority capabilities and limitations which B wishes to grant A. By signing the certificate B explicitly assumes responsibility for A's identity and authority.

The present invention permits B to specify other signators who are required to cosign actions taken by A when using the certification. As noted above, B can further define in the certificate for A the degree to which B will recognize subcertifications performed by A.

Additionally, many other limitations and restrictions may be imposed by B. For example, B may impose a money limit which will insure that any recipient of A's certificate will be aware of the limit B is willing to authorize. Since the process of creating a certificate, as will be shown below involves signatures, the use of cosignatures is extended to permit certification authorization. For example, certificates may be designed to allow delegation of subcertification, but only if particular multiple cosigners are involved. This allows checks and balances to be structured into a hierarchy of authority so that electronic digital signatures can be used across organization and institutional boundaries with great confidence—both by the parties receiving the signatures and the parties granting the authority to use the signatures.

The manner in which a party B creates a certificate for party A is shown in FIG. 5. As indicated at 100, A creates a public/private key pair in accordance with conventional public key signature systems and supplies the public key to B 102. Once B obtains the public key provided by A for certification, it is important for B to insure that the public key is actually one generated by A and not someone masquerading as A. In this regard, it may be desirable for the public key generated by A to be provided on a face to face basis.

Having selected his own certificate with which to sign A's certificate, B at 106 utilizes the certificate 108 with the associated public key 110 to create a signature of a new certificate 112. As in FIG. 2, the signature is created using an object (A's certificate 116) and a certificate (B's certificate 108). B's secret private key is utilized in the decrypt operation to create the signature 112 of the new certificate 116 and the signature packet 114 of B's signature becomes part of A's new certificate packet.

Focussing on the certificate for A which is constructed using information about A specified by B, B builds the certificate by utilizing the public aspect of A's public key as provided by A via line 103. B also sets forth A's full name, A's title and other important statistics such as his address, and telephone number. B may also include a comment to go with the certification which will be available to any person in the future who needs to examine A's certificate.

B additionally will indicate an expiration date of the certificate. This date may reflect the date after which A should not use the certificate. Alternatively, the date may call for any certificate created by A to also expire on this date. B may also indicate in the certificate an account number for A which may represent an internal identification code within B's organization.

Additionally, B may place a monetary limit in the certificate. This monetary authority or credit limit is checked against the limit in B's own certificate to insure that B does not delegate more than he is empowered to delegate. This same relationship is also verified by future recipients as part of their validation process.

As discussed above, B also defines the degree of responsibility to which B is willing is assume for subcertifications done by A. This field must be compatible with the trust level which is allowed B's own certificate. The relationship between the trust level granted to B and that granted by B is another of the relationships validated whenever future recipients validate the hierarchy of certificates which are presented to them.

Finally B inserts cosignature requirements into A's certificate which specify how many and what type of cosignatures are required to accompany A's signature when A uses this new certificate. As indicated above, cosignatures may be in the form of joint signatures and/or counter signatures. The counter signature indicates an approval of the use of the certificate and the approval necessarily follows the associated signature. Joint signatures can be done in any order and do not necessarily reflect approval of the other signatures, but simply approval (or recognition) of a common object.

Cosignature requirements may, for example, be specified in the certificate in a variety of ways. One technique which may be used is to explicitly define a list of valid joint signers and a list of valid counter signers. Associated with each list is the number specifying the minimum associated signatures which must be present in order for a recipient to recognize the signature as being fully approved. The joint signature list may be a vector of hash values of each of the set of other public keys. Some specified minimum number of these keys must appear in certificates of other signatures applied to any object signed by A when using this new certificate. Otherwise any recipient should not treat A's signature as valid.

The counter signature list is a vector of hash values of other certificates which must be used to sign any signature made under the authority of this certificate. Since this references certificates (rather than public keys), it is possible to reference specific certificates which themselves need further joint or counter signing. By select-

ing appropriate certificates to appear here, it is possible to create hierarchy of counter signature requirements to whatever level an organization feels comfortable. A specified number of cosigners is required from each category. This can range from all the candidates to some subset, for example, 0, 1, 2 or 3.

The set of possible co-signers may be indicated explicitly in a list as described here, or implicitly by specifying some quality or attribute specification which is designated in each possible co-signer's certificate.

Other fields may be included in the certificate. For example, the current date and time which reflects the moment of the initial creation of the certificate. As indicated in FIG. 5, the complete certificate consists of a certificate packet with includes the certificate 116 for A and the signature packet 114 of B's signature to A's certificate.

B's signature and the hierarchy of all certificates and signatures which validate it are kept by A and sent along whenever A uses his certificate. It is contemplated that B or other parties may create several certificates for A. For example, one certificate might allow A to reliably identify himself with no further designated authority. Another certificate might allow authorization to A of certain limited money amounts without requiring any cosignatures. A third might allow authorization for larger amounts but require one or more cosignatures. Still another might allow A to subcertify other persons according to still different money and/or authority limitations and/ or co-signature specifications.

Assuming that B has created a certificate for A as shown in FIG. 5, if B requires no cosigners then the certificate is complete. However, the certificate which empowered B to create A's certificate may have required that B have cosigners. There may be one or more joint signature and/or counter signature requirements.

FIG. 6 exemplifies the steps taken by party C to jointly certify the certificate of A. The requirement to have a joint signer would be specified in B's own certificate. In this case, a transmitted object (in this case A's new certificate) signed with B's certificate would be rejected by a recipient if C's joint signature is not also present on the object.

As shown in FIG. 6, if such a joint signature is required, a copy of B's certificate for A is sent to C who must jointly sign the certificate 120. C then examines A's certificate 122 and verifies that the public key of the certificate actually belongs to A in accordance with process outlined in conjunction with FIG. 3.

C then examines the signed attributes and authorizations set forth in the certificate including the assigned monetary level, trust level, etc. C then, upon concluding that all the fields in B's certificate for A are appropriate, selects his own certificate with which to perform the signature 126. With his own certificate 128, C signs B's certificate of A 132 (130). Once C signs his certificate his signature appears essentially parallel with B's signature and any other cosigners as shown at 134 and 136 of FIG. 6. Thus, it is important that C exercise as much caution as B when approving A's certificate. Once A's certificate is created no cosigner may change the certificate for to do so would create essentially a different object to which none of the previous signatures would apply. If C does not approve the certificate he must avoid signing it, and should have a different certificate constructed and re-signed by all necessary parties. After C adds his joint certificate to B's certificate of A, A's certificate packet consists of the certificate for A 132, B's signature packet for A's certificate 134 and finally C's signature packet for A's certificate 136.

In regard to C's signature packet, it is noted that, in order for C to validly sign the certificate, he must select one of his own certificates which grants him sufficient authority to cover what is specified in the new certificate for A. If C has no such certificate, then it is impossible for him to validly sign the certificate since future recipients would reject his certificate as having insufficient authority.

It is noted that C's certificate may also require a counter signature by another party. If so, C forwards the certificate and all associated signatures to the specified party, e.g., D, to counter sign C's signature. When D receives the material he performs the same verification steps as C on the new certificate. If he approves, then D adds his signature to the set. However, D signs C's signature rather then the original certificate object. That is, the object of D's signature is not the object of C's signature (which in this case was the certificate for A) but rather the object is C's signature itself. This counter signature therefore differs from the joint signature which is simply another signature of the same object.

The application of joint and/or counter signatures can be nested to whatever depth is required, Thus, if D is required to have joint signatures, then this package should be passed to one of D's candidate joint signers for approval of C's signature. This would be a joint counter signature. Similarly, in organizational hierarchies it is possible that D might require counter signatures in which case someone else will need to sign D's signature.

As explained above, the recipient of a primary object (such as a purchase order) and its associated signatures, processes the received materials to insure that the object is identical to the material which existed as it was signed by the owner of the public key. The process for verifying the signature and for verifying that the object had not been tampered with has been explained above in regard to FIG. 3.

Additionally, the recipient needs to verify that the identity of the signer is correct and further that the signer has the proper authority within his organization to make the commitments implied by the received object. The sender of the object (e.g., the purchase order) has the responsibility of sending all generations of antecedent certificates and signatures (up to and including the meta-certificate) which are needed for a recipient to perform validation operations.

In validating the object and its signatures, the recipient may, for example proceed as follows. First the recipient insures that the primary object 150 has at least one signature. In the example shown in FIG. 7, the primary object 150 has four associated joint signatures 152, 168, 180 and 200, each of which has associated certificates 154, 170, 182 and 202 respectively.

Certificate 154 was made requiring joint signatures by the owners of certificates 170, 182 and 202, and counter-signatures by the owners of certificates 162 and 166 using these specific certificates. The certificate 154 itself was authorized by the owner of certificate 158 as evidenced by signature 156.

In this example, the owner of 154 has obtained the necessary counter signatures 160 and 164 by the holders

of certificates 162 and 166, as well as the necessary joint-signatures 168, 180 and 200.

To provide validation for his signature 168, the owner of certificate 170 must include the authorization for his certificate. His certificate was signed by the holder of certificate 174 (as evidenced by 172), however 174's certificate specified that a joint signature by the owner of 178 was required in order to fully ratify 174's signature 172. Thus signature 176 which was made sometime in the past, fulfilled all of 174's joint signature requirements and thereby validated (ratified) the use of 170.

Looking at joint signature 180, by the owner of 182, we learn that 182 requires counter signatures by the holder of 186 using the specific certificate 186. The holder of 182, did in fact get the counter-signature 184 by the holder of 186. However, certificate 186 requires that any signature by 186 itself be countersigned by the holders of certificates 190 and 194 (using these respective certificates). These two holders have in fact countersigned 184 as evidenced by 188 and 192. At one further level we learn that certificate 194 requires any signature by 194 be counter signed by the holder of certificate 198, which signature 196 has been obtained. Certificate 202 requires no co-signature.

All certificates must be accompanied by signatures which are themselves authorized by antecedent certificates. Ultimately all the authority can be traced to a set of certificates which have been signed by the holder of the meta-certificate (or possibly a small set of meta-certificates). Each meta-certificate is well known and distributed to all parties "throughout the world".

The recipient examines every signature supplied and verifies that each accurately signs its purported object (whether the object is a primary object, a certificate, or another signature) using the procedure detailed in FIG. 3. The recipient insures that each signature includes a corresponding validated certificate.

If a certificate requires joint signatures, then the recipient insures that the required number of these necessary signatures (to the same object) are present. If the certificate requires counter signatures, then the recipient insures that the required number from the designated subset are present (the counter signatures have signatures as their object).

All certificates are then examined. A check is made for the special meta-certificate which has no signature but which is universally known and trusted and a copy of which is already stored in the recipient's computer. If a certificate is received which claims to be the meta-certificate but which is not equal to that already known to and accepted by the recipient, then a rejection is issued. If the meta-certificate is properly recognized, then the validation process continues.

Additionally, a check is made to insure that any other certificate besides the meta-certificate has at least one signature. As noted above, a check is made to insure that all necessary cosignatures for all presented objects are present. Additionally, a check is made to determine that antecedent certificates grant sufficient authority to the subcertificate signers to permit them to validly sign the certificate.

In this regard, the trust value in the certificate must be consistent with the antecedent (i.e., the certificate of its signers). By way of example only, the following trust field combinations are valid (using the example specified earlier).

Antecedent Trust Value	Trust Value and Immediate Certificate
0	1
0	2
0	3
1	2
1	3
2	2
2	3

Additionally, any monetary limitations set forth in the certificate must be observed. The money limit allowed by a certificate must not exceed its antecedent. Additionally a check should be made to insure that the antecedent's expiration date is compatible with the antecedent's expiration date. By way of example only, a check may be made to insure that the expiration date in every certificate exceeds the date of each signature which relies on it. In some cases, it may be desirable to reject any material which is controlled by an obsolete certificate.

In order to detect "closed" authority loops (by which a series of certificates may be structured in a loop with the last member of the loop granting authority to the first), it is necessary to insure that all authority ultimately flows from recognized meta-certificates. In this manner, a chain of false or artificial certificates which mutually certify each other will not be inadvertently allowed to incorrectly pass the validation process.

One way to accomplish this is to check off certificates in a series of iterations, starting at the top with the meta-certificate. At each iteration, certificates are scanned and in the process certificates having at least one checked off antecedent would in turn be checked off. The iteration stops when no new certificates have been checked off. If any certificates have not been checked off, then these are "orphans" which should never have been supplied. Such a transmitted package would be rejected.

Once the signatures and certificates are validated (based on the ultimate authority of the meta-certificate(s)), the final step is to insure that the commitment inherent in the primary object is within the authority granted to its immediate (joint) signers. This is done by considering the value imputed to the primary object with the certificates of its signers.

Although the use of a meta-certifier insures that all authority ultimately flows from a trusted source and provides protection, the present invention is not limited to a certification methodology which necessarily includes a single meta-certifier. On the other hand, it is contemplated by the present invention to allow for the use of multiple meta-certifiers. These should be certificates governed by entirely independent sources possibly reflecting the apex of entirely different authorizing hierarchies (e.g., the governmental sector versus the private sector).

Another use of multiple meta-certifiers could be to avoid concentrating full meta-certification responsibility with one group. For example, it might be uncomfortable to know that there is a single entity which could in theory create forgeries on behalf of anyone else by creating false certificates. This concern may be alleviated if the meta-certification authority were distributed among different trusted meta-certifiers. Each meta-certifier would operate completely independently but each certificate would specifically require the others as joint

signers. This would essentially eliminate the possibility that isolated corruption within a single organization would compromise the system. For example, any organization that wished to be certified would need to have their own high level master certificate corroborated by each separate entity. Large organizations may likewise wish to structure their own master certificates to be constructed so as to require joint signatures in order to provide multiple safeguards against the danger of isolated corruption within the organization.

While the invention has been described in connection with what is presently considered to be the most practical and preferred embodiment, it is to be understood that the invention is not to be limited to the disclosed embodiment, but on the contrary, is intended to cover various modifications and equivalent arrangements included within the spirit and scope of the appended claims.

I claim:

1. In a communication system having a plurality of terminal devices coupled to a channel over which users of said terminal devices may exchange messages, at least some users having a public key and an associated key, an improved method for managing authority by digitally signing and certifying a message to be transmitted to an independent recipient comprising the steps of:

formulating at least a portion of a digital message;

digitally signing at least said portion of said message; and

including within said message an authorizing digital certificate having a plurality of digital fields created by a certifier, said authorizing certificate being created by the steps of:

specifying by the certifier in at least one of said digital fields, the authority which is vested in the certifier and which has been delegated to the signer of said message, by including sufficient digital information to enable said independent recipient of said message to verify, be electronically analyzing said message in accordance with a predetermined validation algorithm, that the authority exercised by the signer in signing the content of said message created by the signer was properly exercised by the signer in accordance with the authority delegated by the certifier; and

identifying the certifier who has created the signer's certificate in other of said digital fields by including sufficient digital information for said recipient of the message to determine by electronically analyzing said message that the certifier has been granted the authority to grant said delegated authority.

2. A method according to claim 1, further including the step of providing at least one digital field in said message identifying the nature of the digital data being transmitted.

3. A method according to claim 2, wherein the nature of the digital data is identified as being a digital signature.

4. A method according to claim 2, wherein the nature of the digital data is identified as being a certificate.

5. A method according to claim 2, wherein the nature of the digital data is identified as being a business document

6. A method according to claim 1, wherein the formulating step includes the step of providing a digital field allowing the user to insert a predetermined comment regarding the data being transmitted.

7. A method according to claim 1, further including the step of applying a hashing function to at least a portion of the message to be transmitted to form a presignature hash; and wherein said digitally signing step includes the step of processing said presignature hash with the signer's private key to form said digital signature.

8. A method according to claim 7, further including the step of forming a digital signature packet comprising the digital signature and a representation of said at least a portion of the message to be transmitted.

9. A method according to claim 1, wherein said authorizing certificate includes digital fields defining the cosignature requirements which must accompany the signer's signature in order for the signer's signature to be treated as properly authorized.

10. A method according to claim 9, wherein said digital fields defining co-signature requirements set forth a required digital signature by a specified third party indicating approval of the signer's signature to thereby define a counter signature requirement.

11. A method according to claim 10, wherein the third party countersigns by digitally signing the signer,s digital signature.

12. A method according to claim 9, wherein the cosignature requirements include a digital field specifying at least one other digital signature which is required to appear in the digital message thereby defining a joint signature requirement.

13. A method according to claim 1, wherein said authorizing certificate includes at least one digital field defining limitations as to the authority granted by the certificate.

14. A method according to claim 1, wherein said authorizing certificate defines the plurality of the signer.

15. A method according to claim 13, further including the step of specifying a monetary limit for the signer in a digital field in said certificate.

16. A method according to claim 1, wherein said authorizing certificate includes at least one digital field defining a trust level indicative of the degree of responsibility delegated to the signer by the certifier.

17. A method according to claim 1, wherein said identifying step includes the step of specifying in digital fields in said authorizing certificate a hierarchy of certificates, whereby a recipient of the message can electronically verify in accordance with a predetermined validation algorithm the authority of the signer based upon an analysis of the signed message.

18. A method according to claim 1, wherein said step of creating an authorizing certificate includes the steps of creating a certificate by a certifier, whereby the certifier signs the certificate by using the private key associated with one of the certifier's own certificates.

19. A method according to claim 1, including the step of transmitting a plurality of certificates, and wherein at least one of the transmitted certificates is a meta-certificate, where a meta-certificate is a digital authorizing certificate from which authority flows which originates from a trusted source commonly known to both the signer and prospective recipients.

20. In a communications system having a plurality of terminal devices coupled to a communications channel over which users of said terminal devices may exchange messages, at least some of said users having a public key and an associated private key, an improved method of

digitally signing and certifying a message to be transmitted for managing authority comprising the steps of:

formulating at least a portion of a digital message;

digitally signing at least said portion of said message;

including within said message an authorizing digital certificate having a plurality of digital fields created for the signer by a certifier, said authorizing certificate being created by the steps of:

specifying by the certifier in at least one of said digital fields at least one party whose digital signature, in addition to the signer's signature, is required to be transmitted with said message in order for said signer's signature to be treated as properly authorized; and

identifying the certifier who has created the signer,s certificate in other of said digital fields by including sufficient digital information to enable the recipient of said message to determine by electronically analyzing said message that the certifier has been granted the authority to certify the signer's certificate.

21. A method according to claim 20, wherein said certificate includes digital fields representative of a list of each of the public keys of the parties at least one of which is required to cosign any message signed with the authority of the certificate.

22. A method according to claim 20, wherein said certificate includes digital fields representative of a list of public keys of the parties at least one of which may be required to sign any message created under the authority of said certificate and a field defining the minimum member of such signatures which must appear in said message in order for the signer's signature to be treated as properly authorized.

23. A method according to claim 20, wherein said certificate includes digital fields representative of a list of each of the certificates of the parties at least one of which is required to sign any message created under the authority of said certificate.

24. A method according to claim 20, including the step of including digital fields in said message associating with each digital signature in said message an authorizing certificate generated by a certifying party which specifies the authority which has been granted to the message sender.

25. A method according to claim 21, further including the steps of transmitting said message including said certificates and verifying at the recipient,s terminal device each signature through the use of at least one public key.

26. A method according to claim 20, wherein said step of including an authorizing certificate includes the step of defining a hierarchial ladder of certificates within digital fields in the transmitted message, whereby a recipient of the message can electronically verify in accordance with a predetermined validation algorithm the authority of the sender based upon an analysis of the signed message.

27. A method according to claim 20, further including the step of creating an authorizing certificate by a certifier, wherein the certifier creates a certificate by signing the certificate by using the private key associates with one of the certifier's own certificates.

28. A method according to claim 20, further including the step of providing at least one field in said message identifying the nature of the digital data being transmitted.

29. A method according to claim 28, wherein the nature of the digital data is identified as being a digital signature.

30. A method according to claim 28, wherein the nature of the digital data is identified as being a digital certificate.

31. A method according to claim 20, further including the step of applying a hashing function to at least a portion of the message to be transmitted to form a presignature hash; and wherein said digitally signing step includes the step of processing said presignature hash with the signer's private key to form said digital signature.

32. A method according to claim 20, wherein said authorizing certificate includes at least one digital field defining the requirement of at least one digital signature by at least one third party indicating approval of the sender's signature, thereby defining a countersignature requirement, wherein the third party countersigns by digitally signing the sender's digital signature.

33. A method according to claim 20, wherein said authorizing certificate includes at least one digital field specifying at least one additional party required to sign said portion of the digital message to thereby define a joint signature requirement.

34. A method according to claim 20, wherein said authorizing certificate includes at least one digital field defining limitations as to the authority granted by the certificate.

35. A method according to claim 34, wherein said limitations includes a monetary limit for the signer.

36. A method according to claim 20, wherein said authorizing certificate includes at least one digital field indicative of the degree of responsibility delegated to the signer by the certifier.

37. A method according to claim 36, wherein said at least one field defines a trust level indicating the degree of responsibility the certifier is willing to assume for subcertification done by the signer.

38. A method according to claim 20, wherein said authorizing certificate includes at least one field identifying the signer.

39. A method according to claim 20 further including the step of transmitting a plurality of certificates, and wherein at least one of the transmitted certificates is a meta-certificate where a meta-certificate is a digital authorizing certificate from which all authority flows, said meta-certificate originating from a trusted source commonly known to both the signer and the recipient.

40. A method of digitally signing and certifying a sender's message to enable a recipient to determine that the send-r is properly authorized comprising the steps of:

specifying in at least one digital field in an authorizing digital certificate created by a certifier the delegated authority which has been granted to the sender, said authorizing certificate including a plurality of digital fields;

identifying in other of said digital fields in said certificate the identity of the certifier by including sufficient digital information for said recipient to determine that the certifier has been granted the authority to grant the delegated authority;

transmitting a message to said recipient having at least one digital signature, said message including said digital certificate which specifies the authority which has been granted to the sender;

receiving said message by said recipient and validating the identity of the sender by electronically analyzing the at least one digital signature; and

determining the authority which has been granted to the sender by analyzing the delegated authority information specified in said authorizing certificate and determining by electronically analyzing said digital fields that said certifier has been granted the authority to grant said delegated authority.

41. A method according to claim 40, wherein said at least one digital signature is created by computing a presignature hash and said step of validating the identity of the sender including the step of recomputing said presignature hash with the received message,

encrypting the signature to be verified, comparing the recomputed presignature hash and said encrypted signature to be verified; and

rejecting said signature if there is not a match.

42. A method according to claim 41, wherein said encrypting operation is performed with the sender's public encrypting key.

43. A method according to claim 40, further including the step of electronically verifying by a predetermined verification algorithm that the received message is identical to the message as it was initially signed.

44. A method according to claim 40, further including the steps of:

specifying in digital fields in said message at least one digital signature in addition to the signer's signature required to be transmitted;

transmitting said at least one digital signature required to be transmitted and at least one associated certificate;

electronically examining, upon receipt of said message, all received digital certificates and signatures; and

determining in accordance with a predetermined validation algorithm that all necessary signatures are present and that the sender is properly authorized based on data contained in said certificates.

45. A method according to claim 40, wherein said authorizing certificate includes at least one field defining the identity of the signer.

46. A method according to claim 40, further including transmitting a plurality of certificates and wherein at least one of the transmitted certificates is a meta-certificate, where a meta-certificate is a digital authorizing certificate from which authority flows which originates from a trusted source commonly known to both the signer and the recipient.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 4,868,877

DATED : September 19, 1989

INVENTOR(S) : Addison M. Fischer

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 19, line 23, after "some" insert -- of said--;
after "associated" insert -- private--;

Column 19, line 39, delete "be" and insert --by--.

Column 20, line 23, delete "signer,s" and insert --signer's--;

line 35, delete "plurality" and insert --identity--.

Column 21, line 15, delete "signer,s" and insert -- signer's--;

Column 21, line 32, delete "member" and insert --number--;

Column 21, line 49, delete "recipient,s" and insert --recipient's--.

Column 22, line 53, delete "send-r" and insert --sender--.

Signed and Sealed this

Twentieth Day of August, 1991

Attest:

Attesting Officer

HARRY F. MANBECK, JR.

Commissioner of Patents and Trademarks